Accuracy and Enhancement of the Lattice Boltzmann Method for Application to a Cell-Polymer Bioreactor System

A Dissertation
Presented to
The Academic Faculty

By

Marnico D. Deladisma

In Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy in the
George W. Woodruff
School of Mechanical Engineering

Georgia Institute of Technology
May 2006

Accuracy and Enhancement of the Lattice Boltzmann Method for Application to a Cell-Polymer Bioreactor System

Approved By:

Dr. Marc K. Smith, Advisor
Mechanical Engineering
Georgia Institute of Technology

Dr. Cyrus K. Aidun
Mechanical Engineering
Georgia Institute of Technology

Dr. G. Paul Neitzel
Mechanical Engineering
Georgia Institute of Technology

Dr. Fotis Sotiropoulos
Civil Engineering
University of Minnesota

Dr. Suresh Menon
Aerospace Engineering
Georgia Institute of Technology

Date Approved: April 6, 2006

For my wife, Kah Mun

Who can find a virtuous and capable wife?
She is worth more than precious rubies.

Proverbs 31:10

Acknowledgments

There are many people who have assisted in completing this research. First, this work has been supported in part by NASA and the Georgia Tech/Emory Center for the Engineering of Living Tissues (GTEC). I would also like to acknowledge the Woodruff School, particularly Dr. Bill Wepfer, Dr. Yogendra Joshi, and Dr. Wayne Whiteman, for their support as well. A special thanks to my reading committee for their time and consideration throughout this process. I would especially like to thank Dr. Marc Smith, my advisor, for his assistance and support in guiding me through graduate school.

To my parents, Adelfa and Marconi, I would like to say a thank you for many years of unconditional love and support. To my siblings, Marc and Adeline, thank you for always being there for me. To my extended family in Atlanta, including my aunts, cousins, and in-laws, thank you for feeding me all these years! To my friends, thank you for your fellowship and for saying a prayer for me in difficult times. And finally to my wife, Kah Mun, words cannot express what you have meant to me. You have motivated and inspired me to achieve and you have done so all along with a servant's heart.

Lastly, I would like to acknowledge my Lord and Savior Jesus Christ. In 2001, I was fortunate enough to seek you and find you through some hard circumstances. You in turn lifted me out of the mire and set my feet upon rock. Thank you for the growing, personal relationship that we have today. These results do not belong to me, but are for the glory of God the Father in heaven.

Table of Contents

Acknowledgments iv

List of Tables viii

List of Figures xiv

Abbreviations and Symbols xxvi

Summary xxx

Chapter 1 Introduction 1

- 1.1. The RWV Bioreactor 1
- 1.2. Problem Overview 2
- 1.3. Numerical Method 3
- 1.4. Solution Methodology 4
- 1.5. Organization of Thesis 5

Chapter 2 Literature Review 7

- 2.1. Bioreactors 7
  - 2.1.1. Overview 8
  - 2.1.2. RWV Bioreactor Studies 9
  - 2.1.3. Cells and their Response to Shear Stresses 12
  - 2.1.4. Characterization of Fluid Dynamics in Bioreactors 14
- 2.2. The Lattice Boltzmann Method 18
  - 2.2.1. Numerical Methods and Commercial Solvers 19
  - 2.2.2. Overview of the Lattice Boltzmann Method 21
  - 2.2.3. Lattice Boltzmann Applications 23
  - 2.2.4. Lattice Boltzmann Models 24
  - 2.2.5. Boundary Treatments 32
  - 2.2.6. Particle Dynamics 46
  - 2.2.7. Grid Refinement 56
  - 2.2.8. Accuracy 63
- 2.3. Summary 68

Chapter 3 Methodology 69

- 3.1. Implementation of the Lattice Boltzmann Method 69
  - 3.1.1. 2D and 3D Models 70
  - 3.1.2. Boundary Treatment 73
  - 3.1.3. Particle Dynamics 82
  - 3.1.4. Grid Refinement 94
  - 3.1.5. Shear Stresses 99

3.1.6. Implementation Issues 103
3.2. Programming Software 110
3.2.1. Visual C++ 6.0 112
3.2.2. MATLAB 114
3.3. Parallel Implementation 117
3.3.1. PVM 119
3.3.2. NT-MPICH 123
3.4. Summary 126

Chapter 4 Numerical Studies, Evaluation, and Results 127

4.1. Numerical Studies 128
4.1.1. Boundary Treatment Studies 129
4.1.2. Grid Refinement Studies 137
4.1.3. Dynamics Studies 142
4.1.4. Simulations 149
4.2. Evaluation 151
4.2.1. Software Methodology 152
4.2.2. Analysis Methodology 156
4.3. 2D Results 160
4.3.1. Boundary Treatment Studies 161
4.3.2. Grid Refinement Studies 175
4.3.3. Dynamics Studies 188
4.3.4. Simulations 209
4.4. 3D Results 212
4.4.1. Boundary Treatment Studies 213
4.4.2. Grid Refinement Studies 228
4.4.3. Dynamics Studies 237
4.4.4. Simulations 249
4.5. Summary 252

Chapter 5 Analysis and Discussion 254

5.1. Accuracy of the Lattice Boltzmann 254
5.1.1. Boundary Treatment 255
5.1.2. Grid Refinement 269
5.1.3. Dynamics 278
5.2. Application to the RWV Bioreactor 290
5.2.1. Implementation 291
5.2.2. Simulations 292
5.3. Summary 297

Chapter 6 Contributions and Recommendations 298

6.1. Contributions of this Research 298
6.2. Recommendations for Future Work 303

Appendix A – Quaterion Math 305

Appendix B – Partial Derivative Calculation 308

Appendix C – 2D Results 310

Appendix D – 3D Results 383

References 450

Vita 458

List of Tables

Table 4.1	Lattice Boltzmann Parameters for the 2D HCF Grids	162
Table 4.2	Lattice Boltzmann Parameters for the 2D LDCF Grids	163
Table 4.3	Velocity and Viscosity Values used in 2D Annulus Cases	167
Table 4.4	Lattice Boltzmann Parameters for the 2D Annulus Grids	168
Table 4.5	Stability of Lattice Boltzmann 2D Annulus Results	169
Table 4.6	Grid Refinement Computations for the 2D LDCF Main Grid Resolutions	177
Table 4.7	Grid Refinement Computations for the 2D ANN Main Grid Resolutions	183
Table 4.8	Lattice Boltzmann Parameters for the 2D Lubrication Grids	198
Table 4.9	Lattice Boltzmann Parameters for the 2D VCF Grids	201
Table 4.10	Lattice Boltzmann Parameters for the 2D Simulation	209
Table 4.11	Lattice Boltzmann Parameters for the 3D HCF Grids	215
Table 4.12	Lattice Boltzmann Parameters for the 3D LDCF Grids	216
Table 4.13	Lattice Boltzmann Parameters for the 3D Annulus Grids	221
Table 4.14	Stability of Lattice Boltzmann 3DAnnulus Results	222
Table 4.15	Grid Refinement Computations for the 3D LDCF Main Grid Resolutions	229
Table 4.16	Grid Refinement Computations for the 3D ANN Main Grid Resolutions	234
Table 4.17	Lattice Boltzmann Parameters for the 3D Lubrication Grids	244
Table 4.18	Lattice Boltzmann Parameters for the 3D VCF Grids	247
Table 4.19	Lattice Boltzmann Parameters for the 3D Simulation	249

Table 5.1	Comparison of Boundary Treatments	256
Table 5.2	X-Velocity Percent Error Results for the ANN3 Cases	258
Table 5.3	X-Velocity RMS Error Results for the ANN3 Cases	258
Table 5.4	Velocity RMS Error Results for the 2D Annulus Cases	260
Table 5.5	Percent Error Results for the 2D Annulus Cases	260
Table 5.6	Velocity RMS Error Results for the 3D Annulus Cases	261
Table 5.7	Percent Error Results for the 3D Annulus Cases	261
Table 5.8	Average Force Percent Error Results for the 2D Annulus Cases	262
Table 5.9	Percent Decrease in X-Velocity RMS Error for 2D Cases	271
Table 5.10	Percent Decrease in Y-Velocity RMS Error for 3D Cases	272
Table 5.11	Y-Velocity RMS Error Comparison for ANN-CIR Sub-Grid Cases	274
Table 5.12	Force Percent Error Results Comparing 2D Main Grid and Sub-Grid Cases	276
Table 5.13	Force Percent Error Results for 2D Annulus Force Evaluation Cases	279
Table 5.14	Force Percent Error Results for 2D Annulus Correction Cases	281
Table 5.15	Force Percent Error Results for 2D Annulus Cases	282
Table 5.16	Force Percent Error Results for 3D Annulus Cases	283
Table 5.17	Lattice Boltzmann Parameters for the Simulations	292
Table C.1	Channel Resolution (L) versus RMS Error for all Cases	313
Table C.2	X-Velocity Percent Error Results for the LDCF Cases	325
Table C.3	Y-Velocity Percent Error Results for the LDCF Cases	325
Table C.4	X-Velocity Percent Error Results for the ANN-CAS5 Cases	329
Table C.5	X-Velocity Percent Error Results for the ANN-CAS6 Cases	329

Table C.6	X-Velocity Percent Error Results for the ANN-CAS8 Cases	330
Table C.7	Y-Velocity Percent Error Results for the ANN3-CAS5 Construct Cases	333
Table C.8	Y-Velocity Percent Error Results for the ANN-CIR-CAS5 Cases	334
Table C.9	Y-Velocity Percent Error Results for the ANN3-EL Flow Parameter Cases	334
Table C.10	RMS Error Results for the CAS4-ANN2 Density Cases	336
Table C.11	Percent Decrease in X-Velocity RMS Error for LDCF-UL Cases	340
Table C.12	Percent Decrease in Y-Velocity RMS Error for LDCF-UR Cases	340
Table C.13	X-Velocity Percent Error Results for the LDCF-UL Sub-Grid Cases	341
Table C.14	X-Velocity Percent Error Results for the LDCF-UL SGA Cases	341
Table C.15	Y-Velocity Percent Error Results for the LDCF-UR Sub-Grid Cases	342
Table C.16	Y-Velocity Percent Error Results for the LDCF-UR SGA Cases	342
Table C.17	RMS X-Velocity Error Results for LDCF-LM Cases	343
Table C.18	Percent Decrease in Y-Velocity RMS Error for ANN-CIR Cases	347
Table C.19	Percent Decrease in X-Velocity RMS Error for ANN-SQ Cases	347
Table C.20	Y-Velocity Percent Error Results for the ANN-CIR Cases	348
Table C.21	Y-Velocity Percent Error Results for the ANN-CIR SGA Cases	348
Table C.22	X-Velocity Percent Error Results for the ANN-SQ Cases	349
Table C.23	X-Velocity Percent Error Results for the ANN-SQ SGA Cases	349

Table C.24	Percent Decrease in Y-Velocity RMS Error for ANN-SQ-RF4 Flow Parameter Cases	350
Table C.25	X-Velocity Percent Error Results for the ANN-SQ-RF4 Flow Parameter Cases	350
Table C.26	X-Velocity Percent Error Results for the ANN-SQ-RF4 SGA Flow Parameter Cases	351
Table C.27	Percent Decrease in X-Velocity Error for the ANN3-EL Mix Cases	352
Table C.28	Percent Decrease in Y-Velocity Error for the ANN3-EL Mix Cases	352
Table C.29	Force Results for the ANN-SQ Main Grid Cases	353
Table C.30	Force Results for the ANN-CIR Main Grid Cases	353
Table C.31	Force Results for the ANN-RECT Main Grid Cases	354
Table C.32	Force Results for the ANN-EL Main Grid Cases	354
Table C.33	Force Results for the ANN-SQ Sub-Grid Cases	355
Table C.34	Force Results for the ANN-CIR Sub-Grid Cases	355
Table C.35	Force Results for the ANN-RECT Sub-Grid Cases	356
Table C.36	Force Results for the ANN-EL Sub-Grid Cases	356
Table C.37	Force Results for the ANN3-EL-RF4 Sub-Grid Flow Parameter Cases	357
Table C.38	Force Results for the ANN3-EL-RF4 Sub-Grid Mix Cases	357
Table D.1	Channel Resolution (L) versus RMS Error for All Cases	385
Table D.2	X-Velocity Percent Error Results for the Z = 2.4dm LDCF Cases	395
Table D.3	Y-Velocity Percent Error Results for the Z = 2.4dm LDCF Cases	396
Table D.4	X-Velocity Percent Error Results for the Z = 1.2dm LDCF Cases	396
Table D.5	Y-Velocity Percent Error Results for the Z = 1.2dm LDCF Cases	396

Table D.6 Z-Velocity Percent Error Results for the Z = 1.2dm LDCF Cases 397

Table D.7 X-Velocity Percent Error Results for the 3D ANN-CAS5 Cases 405

Table D.8 Z-Velocity Percent Error Results for the 3D ANN-CAS5 Cases 406

Table D.9 Y-Velocity Percent Error Results for the 3D ANN3-CAS5 Construct Cases 409

Table D.10 Y-Velocity Percent Error Results for the 3D ANN-SPH-CAS5 Cases 409

Table D.11 Z-Velocity Percent Error Results for the 3D ANN-SPH-CAS5 Cases 409

Table D.12 Percent Decrease in X-Velocity RMS Error for 3D LDCF-UL Cases 414

Table D.13 Percent Decrease in Y-Velocity RMS Error for 3D LDCF-UR Cases 414

Table D.14 Percent Decrease in Z-Velocity RMS Error for 3D LDCF-UR Cases 414

Table D.15 X-Velocity Percent Error Results for the 3D LDCF-UL Sub-Grid Cases 415

Table D.16 X-Velocity Percent Error Results for the 3D LDCF-UL SGA Cases 415

Table D.17 Y-Velocity Percent Error Results for the 3D LDCF-UR Sub-Grid Cases 415

Table D.18 Y-Velocity Percent Error Results for the 3D LDCF-UR SGA Cases 416

Table D.19 Z-Velocity Percent Error Results for the 3D LDCF-UR Sub-Grid Cases 416

Table D.20 Z-Velocity Percent Error Results for the 3D LDCF-UR SGA Cases 416

Table D.21 RMS X-Velocity Error Results for 3D LDCF-LM Cases 417

Table D.22 RMS Z-Velocity Error Results for 3D LDCF-LM Cases 417

Table D.23	Percent Decrease in Y-Velocity RMS Error for 3D ANN-CUBE Cases	422
Table D.24	Percent Decrease in X-Velocity RMS Error for 3D ANN-SPH Cases	422
Table D.25	Percent Decrease in Z-Velocity RMS Error for 3D ANN-SPH Cases	422
Table D.26	Y-Velocity Percent Error Results for the 3D ANN-CUBE Cases	423
Table D.27	Y-Velocity Percent Error Results for the 3D ANN-CUBE SGA Cases	423
Table D.28	X-Velocity Percent Error Results for the 3D ANN-SPH Cases	423
Table D.29	X-Velocity Percent Error Results for the 3D ANN-SPH SGA Cases	424
Table D.30	Z-Velocity Percent Error Results for the 3D ANN-SPH Cases	424
Table D.31	Z-Velocity Percent Error Results for the 3D ANN-SPH SGA Cases	424
Table D.32	Force Results for the 3D ANN-CUBE Main Grid Cases	425
Table D.33	Force Results for the 3D ANN-SPH Main Grid Cases	425
Table D.34	Force Results for the 3D ANN-DSC Main Grid Cases	426
Table D.35	Force Results for the 3D ANN-EL Main Grid Cases	426
Table D.36	Force Results for the 3D ANN-CUBE Sub-Grid Cases	427
Table D.37	Force Results for the 3D ANN-SPH Sub-Grid Cases	427
Table D.38	Force Results for the 3D ANN-DSC Sub-Grid Cases	428
Table D.39	Force Results for the 3D ANN-EL Sub-Grid Cases	428

List of Figures

Figure 2.1	The RWV Bioreactor	9
Figure 2.2	Construct Behavior in the RWV Bioreactor	15
Figure 2.3	Force Balance for a Construct in the RWV Bioreactor	16
Figure 2.4	A Typical Time Step in the Lattice Boltzmann Method	27
Figure 2.5	Velocities for the D2Q9 Model	28
Figure 2.6	Illustration of Velocities in the D2Q9 Model	28
Figure 2.7	Illustration of Velocities in the D3Q19 Model	30
Figure 2.8	Illustration of the Bounce-Back and Shifted Bounce-Back Treatments	33
Figure 2.9	Illustration for the Hydrodynamic Boundary Treatment	35
Figure 2.10	Illustration for the MLS Boundary Treatment	39
Figure 2.11	Illustration for the CBB Boundary Treatment	40
Figure 2.12	Illustration for the Chen Extrapolation Boundary Treatment	42
Figure 2.13	Illustration for the Bouzidi Interpolation Boundary Treatment	43
Figure 2.14	Illustration for the Yu Interpolation Boundary Treatment	45
Figure 2.15	Illustration of Momentum Exchange at a Moving Wall	47
Figure 2.16	Illustration of Momentum-Exchange Method for Force Evaluation (fluid exists on both sides of wall)	48
Figure 2.17	Illustration of Momentum-Exchange Method for Force Evaluation (fluid exists only outside of wall)	50
Figure 2.18	Illustration of One Type of Non-Uniform Grid	57
Figure 2.19	Illustration of an Embedded Sub-Grid	58
Figure 2.20	Illustration of Time Steps for a Sub-Grid Implementation	61

Figure 3.1	Velocity Designations for the D2Q9 Model	70
Figure 3.2	Velocity Designations for the D3Q19 Model	71
Figure 3.3	Illustration for Calculation a PDF Value at the Wall	75
Figure 3.4	Illustration for Calculation Position of Bounce-Back Particles	76
Figure 3.5	Illustration for an MDI Boundary Treatment Example in the D2Q9 Model	77
Figure 3.6	Particle Positions for the $\mathbf{e_5}$ Velocity Case	78
Figure 3.7	Particle Positions for the $\mathbf{e_6}$ Velocity Case	80
Figure 3.8	Illustration for Modified Force Evaluation at a Surface	84
Figure 3.9	Illustration of a Moving Boundary in the D2Q9 Model	86
Figure 3.10	Illustration of Body and World Space Coordinates	90
Figure 3.11	Illustration for $\mathbf{I_{body}}$ Calculation	91
Figure 3.12	Illustration for a Non-Uniform Coarsened Grid	95
Figure 3.13	Illustration of a Sub-Grid with N = 3 in the D2Q9 Model	96
Figure 3.14	Timeline for Sub-Grid with N = 3	98
Figure 3.15	Illustration of Grid Layout for Channel Flow in the D2Q9 Model	103
Figure 3.16	Illustration of Grid Layout for 2D Annulus Case	105
Figure 3.17	GUI for the Lattice Boltzmann Software	113
Figure 3.18	Illustration of the MATLAB Surface Fitting with griddata (top view)	116
Figure 3.19	Illustration of the MATLAB Surface Fitting with griddata (isometric view)	117
Figure 3.20	Two Common Parallel Computation Paradigms	120
Figure 3.21	PVM Implementation of Lattice Boltzmann Software	122
Figure 3.22	NT-MPICH Implementation of Lattice Boltzmann Software	125

Figure 4.1	2D Channel Flow Geometry	130
Figure 4.2	Cross Section of Shifted Channel for 3D Flow	132
Figure 4.3	2D Lid-Driven Cavity Flow Geometry	134
Figure 4.4	2D Annulus Case Geometry	136
Figure 4.5	2D Irregular Grid Channel Flow	139
Figure 4.6	2D Coarsened Grid for Lid-Driven Cavity Flow	139
Figure 4.7	Grid Refinement for the 2D Lid-Driven Cavity Flow Case	140
Figure 4.8	Grid Refinement for the 2D Annulus Case	141
Figure 4.9	Approaching Circular Cylinders in a 2D Channel	145
Figure 4.10	2D Sedimentation of a Particle in a Vertical Channel	146
Figure 4.11	2D Sedimentation of Two Particles in a Vertical Channel	147
Figure 4.12	Fieldview Screenshot	153
Figure 4.13	FLUENT Screenshot	154
Figure 4.14	2D Grid Comparison	155
Figure 4.15	Velocity Vectors and Mesh for Grid Refinement in the LDCF UL Case	179
Figure 4.16	Velocity Vectors and Mesh for Grid Refinement in the ANN Case	185
Figure 4.17	Illustration of Line Plot at Y = 1.8dm	194
Figure 4.18	Illustration of a Moving Sub-Grid in the Annulus Geometry	207
Figure 4.19	Illustration of Z = 1.2dm Plane for the LDCF Geometry	218
Figure 4.20	Illustration of Z = 0.27dm Plane for the 3D Annulus Geometry	223
Figure 4.21	Illustration of Z = 0.81dm Plane for the 3D Annulus Geometry	226
Figure 4.22	Illustration of Line Plot at Y = 0.9dm, Z = 0.9dm	241

Figure 5.1	Shear Stress Plot for the 2D Ellipse comparing Boundary Treatments	263
Figure 5.2	Shear Rate Plot for the 2D Ellipse comparing Boundary Treatments	264
Figure 5.3	XZ Shear Stress Plot for the 3D Disc comparing Boundary Treatments	264
Figure 5.4	Shear Rate Plot for the 3D Disc comparing Boundary Treatments	265
Figure 5.5	C versus L/d comparing Boundary Treatments	266
Figure 5.6	Particle Force versus Position for the VCF3 Grid	267
Figure 5.7	Circle Particle Path for ANN3 Case	268
Figure 5.8	Absolute X-Velocity Error for the LDCF4 Main Grid and Sub-Grid	271
Figure 5.9	Shear Rate Plot comparing Main Grid and Sub-Grid Values	273
Figure 5.10	Shear Stress Plot comparing Main Grid and Sub-Grid Values	273
Figure 5.11	Shear Stress Plot comparing results across Main Grids	275
Figure 5.12	Circle Particle Path for ANN0 Case	277
Figure 5.13	Force Evaluation Area Illustration	280
Figure 5.14	XY Shear Stress Plot Illustrating 3D FLUENT Solution	284
Figure 5.15	Shear Rate Contour Plots for the 3D ANN-EL Construct Case	285
Figure 5.16	C versus L/d with Grid Refinement for the BZ Boundary Treatment	287
Figure 5.17	E versus d/L with Grid Refinement for the BZ Boundary Treatment	287
Figure 5.18	Typical Result for Two Particle Sedimentation in a Channel	288
Figure 5.19	Sphere Particle Path for 3D ANN3 Rigid Body Rotation Case	289
Figure 5.20	Velocity Vector Comparison for the 3D Flow around a DSC Construct	294

Figure 5.21 Line Plot at Y = -0.375cm for Velocity Magnitude Comparison between PIV and LB data 295

Figure 5.22 DSC Construct Path for 14RPM Case 296

Figure A.1 Quaternion Notation 305

Figure A.2 Quaternion Multiplication 306

Figure A.3 Quaternion to Matrix Conversion 306

Figure A.4 Matrix to Quaternion Conversion 307

Figure C.1 RMS Error versus $1/L^2$ for the D1 case 310

Figure C.2 RMS Error versus $1/L^2$ for the D2 case 311

Figure C.3 RMS Error versus $1/L^2$ for the D3 case 311

Figure C.4 RMS Error versus $1/L^2$ for the D4 case 312

Figure C.5 Velocity Profile for the HCF1 Grid where L = 0.3m 313

Figure C.6 Streamlines for the FLUENT results for LDCF 314

Figure C.7 Streamlines for the Lattice Boltzmann LDCF1 Case 314

Figure C.8 Streamlines for the Lattice Boltzmann LDCF4 Case 315

Figure C.9 LDCF Velocity Plots using the SBB Treatment 316

Figure C.10 LDCF Velocity Plots using the MDI Treatment 317

Figure C.11 LDCF Velocity Plots using the BZ Treatment 318

Figure C.12 LDCF Velocity Plots using the YU Treatment 319

Figure C.13 X-Velocity Error for the LDCF1 case using the SBB Treatment 320

Figure C.14 Y-Velocity Error for the LDCF1 case using SBB Treatment 320

Figure C.15 X-Velocity Error for the LDCF4 case using the SBB Treatment 321

Figure C.16 Y-Velocity Error for the LDCF4 case using the SBB Treatment 321

Figure C.17 X-Velocity Error for the LDCF4 case using the MDI Treatment 322

Figure C.18	X-Velocity Error for the LDCF4 case using the BZ Treatment	322
Figure C.19	X-Velocity Error for the LDCF4 case using the YU Treatment	323
Figure C.20	X-Velocity RMS Error for the LDCF Case	324
Figure C.21	Y-Velocity RMS Error for the LDCF Case	324
Figure C.22	Typical X-Velocity Error for Taylor-Couette Case	326
Figure C.23	Typical Y-Velocity Error for Taylor-Couette Case	326
Figure C.24	X-Velocity RMS Error for the ANN3 grid using the SBB Treatment	327
Figure C.25	X-Velocity RMS Error for the ANN3 grid using the MDI Treatment	327
Figure C.26	X-Velocity RMS Error for the ANN3 grid using the BZ Treatment	328
Figure C.27	X-Velocity RMS Error for the ANN3 grid using the YU Treatment	328
Figure C.28	Streamline Comparison of FLUENT and LB Solutions for the ANN-RECT Case	330
Figure C.29	Typical X-Velocity Error for Annulus Construct Case	331
Figure C.30	Typical Y-Velocity Error for Annulus Construct Case	331
Figure C.31	Y-Velocity RMS Error versus Construct for ANN3-CAS5 Cases	332
Figure C.32	Y-Velocity RMS Error versus Resolution for ANN-CAS5-CIR Cases	332
Figure C.33	Y-Velocity RMS Error versus Flow Parameters for ANN3-EL Cases	333
Figure C.34	Y-Velocity Absolute Error for Annulus Local Density Case	335
Figure C.35	Y-Velocity Absolute Error for Annulus Average Density Case	335
Figure C.36	Absolute X-Velocity Error for the LDCF4-RF2-UL Case	337
Figure C.37	Absolute X-Velocity Error for the LDCF4 Case	337

Figure C.38	X-Velocity RMS Error for the LDCF1-UL Grid Refinement	338
Figure C.39	X-Velocity RMS Error for the LDCF4-UL Grid Refinement	338
Figure C.40	Y-Velocity RMS Error for the LDCF1-UR Grid Refinement	339
Figure C.41	Y-Velocity RMS Error for the LDCF4-UR Grid Refinement	339
Figure C.42	Y-Velocity Error for the ANN2-RF3-CIR Case	344
Figure C.43	Absolute Y-Velocity Error for the ANN2-CIR Case	344
Figure C.44	Y-Velocity RMS Error for the ANN1-CIR Grid Refinement	345
Figure C.45	Y-Velocity RMS Error for the ANN3-CIR Grid Refinement	345
Figure C.46	X-Velocity RMS Error for the ANN1-SQ Grid Refinement	346
Figure C.47	X-Velocity RMS Error for the ANN3-SQ Grid Refinement	346
Figure C.48	Shear Stress Contour Plots for the ANN-CIR Construct Case	358
Figure C.49	Shear Rate Contour Plots for the Annulus Case with an EL Construct	359
Figure C.50	Shear Stress Plot comparing Main Grid and Sub-Grid Values	360
Figure C.51	Shear Rate Plot Comparing Main Grid and Sub-Grid Values	360
Figure C.52	Shear Stress Plot Comparing Refinement Factors	361
Figure C.53	Shear Rate Plot Comparing Refinement Factors	361
Figure C.54	Shear Stress Plot Comparing Results across Main Grids	362
Figure C.55	Shear Rate Plot Comparing Results across Main Grids	362
Figure C.56	Shear Stress Plot Comparing Results across Boundary Treatments	363
Figure C.57	Shear Rate Plot Comparing Results across Boundary Treatments	363
Figure C.58	Force versus Gap Size for the SBB Boundary Treatment	364
Figure C.59	Force versus Gap Size for the BZ Boundary Treatment	365
Figure C.60	Force versus Gap Size for the No Lubrication Force Cases	366

Figure C.61	Force versus Gap Size for the RF8 Sub-Grid Case	367
Figure C.62	C versus L/d for the Stokes Boundary Condition (SBC)	368
Figure C.63	C versus L/d for the Cavity Boundary Condition (CBC)	369
Figure C.64	C versus L/d with Grid Refinement for the BZ Boundary Treatment	370
Figure C.65	Particle Velocity versus Position for the VCF3 Grid	371
Figure C.66	Particle Force versus Position for the VCF3 Grid	371
Figure C.67	Particle Velocity versus Position for the VCF3-RF Grids	372
Figure C.68	Particle Force versus Position for the VCF3-RF Grid	372
Figure C.69	Typical Result for Two Particle Sedimentation in a Channel	373
Figure C.70	Two Particle Sedimentation Comparison SBB versus BZ	374
Figure C.71	Two Particle Sedimentation Comparison for Net Body Forces	375
Figure C.72	Circle Particle Path for ANN3 Case	376
Figure C.73	Circle Particle Path for ANN0 Case	377
Figure C.74	2D Velocity Vectors for Flow around a RECT Construct	378
Figure C.75	RECT Construct Path for Stationary Case	379
Figure C.76	RECT Construct Path for Orbiting Case	380
Figure C.77	RECT Construct Path for Oscillation Case	381
Figure C.78	XY Shear Stress Contours for the RECT Construct	382
Figure D.1	RMS Error versus $1/L^2$ for the 3D D1 case	383
Figure D.2	RMS Error versus $1/L^2$ for the 3D D2 case	384
Figure D.3	RMS Error versus $1/L^2$ for the 3D D3 Case	384
Figure D.4	RMS Error versus $1/L^2$ for the 3D D4 Case	385
Figure D.5	Streamlines for the 3D FLUENT LDCF at Z = 2.4dm	386

Figure D.6	Streamlines for the 3D LB LDCF at Z = 2.4dm	386
Figure D.7	3D LDCF Velocity Centerline Plots using the SBB Treatment	387
Figure D.8	3D LDCF Velocity Centerline Plots using the BZ Treatment	388
Figure D.9	3D LDCF Velocity Centerline Plots using the YU Treatment	389
Figure D.10	3D LDCF Z-Velocity Plot using the SBB Treatment	390
Figure D.11	3D LDCF Z-Velocity Plot using the BZ Treatment	390
Figure D.12	3D LDCF Z-Velocity Plot using the YU Treatment	391
Figure D.13	Y-Velocity Error for the LDCF3 case using the SBB Treatment	391
Figure D.14	Y-Velocity Error for the LDCF3 case using the BZ Treatment	392
Figure D.15	Y-Velocity Error for the LDCF3 case using the YU Treatment	392
Figure D.16	X-Velocity RMS Error for the 3D LDCF Cases at Z = 2.4dm	393
Figure D.17	Y-Velocity RMS Error for the 3D LDCF Cases at Z = 2.4dm	393
Figure D.18	X-Velocity RMS Error for the 3D LDCF Cases at Z = 1.2dm	394
Figure D.19	Y-Velocity RMS Error for the 3D LDCF Cases at Z = 1.2dm	394
Figure D.20	Z-Velocity RMS Error for the 3D LDCF Cases at Z = 1.2dm	395
Figure D.21	Typical X-Velocity Error for Z = 0.9dm Plane	398
Figure D.22	Typical X-Velocity Error for Z = 0.27dm Plane	398
Figure D.23	Typical Z-Velocity Error for Z = 0.27dm Plane	399
Figure D.24	3D ANN Y-Velocity Plot using the SBB Treatment	399
Figure D.25	3D ANN Z-Velocity Plot using the SBB Treatment	400
Figure D.26	3D ANN Y-Velocity Plot using the BZ Treatment	400
Figure D.27	3D ANN Z-Velocity Plot using the BZ Treatment	401
Figure D.28	3D ANN Y-Velocity Plot using the YU Treatment	401

Figure D.29 3D ANN Z-Velocity Plot using the YU Treatment 402

Figure D.30 X-Velocity RMS Error for the 3D ANN3 grid using the SBB Treatment 402

Figure D.31 X-Velocity RMS Error for the ANN3 grid using the BZ Treatment 403

Figure D.32 X-Velocity RMS Error for the ANN3 grid using the YU Treatment 403

Figure D.33 Z-Velocity RMS Error for the ANN3 grid using the SBB Treatment 404

Figure D.34 Z-Velocity RMS Error for the ANN3 grid using the BZ Treatment 404

Figure D.35 Z-Velocity RMS Error for the ANN3 grid using the YU Treatment 405

Figure D.36 Typical Y-Velocity Error for 3D Annulus Construct Case 406

Figure D.37 Typical Z-Velocity Error for 3D Annulus Construct Case 407

Figure D.38 Y-Velocity RMS Error versus Construct for ANN3-CAS5 Cases 407

Figure D.39 Y-Velocity RMS Error versus Resolution for ANN-SPH-CAS5 Cases 408

Figure D.40 Z-Velocity RMS Error versus Resolution for ANN-SPH-CAS5 Cases 408

Figure D.41 Absolute X-Velocity Error for the LDCF3-RF4-UL Case 410

Figure D.42 Absolute X-Velocity Error for the LDCF3 Case 410

Figure D.43 X-Velocity RMS Error for the LDCF1-UL Grid Refinement 411

Figure D.44 X-Velocity RMS Error for the LDCF3-UL Grid Refinement 411

Figure D.45 Y-Velocity RMS Error for the LDCF1-UR Grid Refinement 412

Figure D.46 Y-Velocity RMS Error for the LDCF3-UR Grid Refinement 412

Figure D.47 Z-Velocity RMS Error for the LDCF1-UR Grid Refinement 413

Figure D.48 Z-Velocity RMS Error for the LDCF3-UR Grid Refinement 413

Figure D.49 Absolute Y-Velocity Error for the ANN3-RF3-CUBE Case 418

Figure D.50 Absolute Y-Velocity Error for the ANN3-CUBE Case 418

Figure D.51 Y-Velocity RMS Error for the ANN1-CUBE Grid Refinement 419

Figure D.52 Y-Velocity RMS Error for the ANN3-CUBE Grid Refinement 419

Figure D.53 X-Velocity RMS Error for the ANN1-SPH Grid Refinement 420

Figure D.54 X-Velocity RMS Error for the ANN3-SPH Grid Refinement 420

Figure D.55 Z-Velocity RMS Error for the ANN1-SPH Grid Refinement 421

Figure D.56 Z-Velocity RMS Error for the ANN3-SPH Grid Refinement 421

Figure D.57 FLUENT Shear Stress Contour Plots for the ANN-EL Construct Case 429

Figure D.58 LB Shear Stress Contour Plots for the ANN-EL Construct Case 430

Figure D.59 FLUENT Shear Rate Contour Plots for the ANN-EL Construct Case 431

Figure D.60 LB Shear Rate Contour Plots for the ANN-EL Construct Case 432

Figure D.61 XY Shear Stress Plot Comparing Main Grid and Sub-Grid Values 433

Figure D.62 XZ Shear Stress Plot Comparing Main Grid and Sub-Grid Values 433

Figure D.63 YZ Shear Stress Plot Comparing Main Grid and Sub-Grid Values 434

Figure D.64 Shear Rate Plot Comparing Main Grid and Sub-Grid Values 434

Figure D.65 XY Shear Stress Plot Comparing Results across Main Grids 435

Figure D.66 Shear Rate Plot Comparing Results across Main Grids 435

Figure D.67 XZ Shear Stress Plot Comparing Results across Boundary Treatments 436

Figure D.68 Shear Rate Plot Comparing Results across Boundary Treatments 436

Figure D.69 Force versus Gap Size for the SBB Boundary Treatment 437

Figure D.70 Force versus Gap Size for the BZ Boundary Treatment 438

Figure D.71 E versus d/L for all Boundary Treatments 439

Figure D.72 E versus d/L with Grid Refinement for the BZ Boundary Treatment 440

Figure D.73 Particle Velocity versus Position for the 3D VCF3 Grid 441

Figure D.74 Particle Force versus Position for the 3D VCF3 Grid 441

Figure D.75 Particle Velocity versus Position for the 3D VCF3-RF Grids 442

Figure D.76 Particle Force versus Position for the 3D VCF3-RF Grid 442

Figure D.77 Sphere Particle Path for 3D ANN3 Case 443

Figure D.78 Velocity Vectors for 3D Flow around a DSC Construct 444

Figure D.79 DSC Construct Path for Stationary Case 445

Figure D.80 DSC Construct Path for Orbiting Annulus Case 446

Figure D.81 DSC Construct Path for 13RPM Case 447

Figure D.82 DSC Construct Path for 14RPM Case 448

Figure D.83 XY Shear Stress Contours for the DSC Construct 449

Abbreviations and Symbols

AC	Articular Cartilage
ANN	Annulus Geometry
BZ	Bouzidi Quadratic Interpolation Boundary Treatment
CAS	Lattice Boltzmann Flow Parameter Case
CBB	Continuous Bounce-Back Boundary Treatment
CFD	Computational Fluid Dynamics
CIR	Circle 2D Construct
CUBE	Cube 3D Construct
DSC	Disc 3D Construct
D2Q9	2D Lattice Boltzmann Model using 9 Velocities
D3Q19	3D Lattice Boltzmann Model using 19 Velocities
ECM	Extracellular Matrix
EL	Ellipse 2D Construct or Ellipsoid 3D Construct
EPDF	Equilibrium Particle Distribution Function
FH	Filippoval, Hänel Boundary Treatment
GAG	Glycosaminoglycan
HCF	Horizontal Channel Flow Geometry
LB	Lattice Boltzmann
LDCF	Lid-Driven Cavity Flow Geometry
LGA	Lattice Gas Automata
LM	Lower Middle of the LDCF Geometry

MDI	Multi-Dimensional Interpolation Boundary Treatment
MLS	Mei, Luo, Shyy Boundary Treatment
MPI	Message Passing Interface
PDF	Particle Distribution Function
PVM	Parallel Virtual Machine
RECT	Rectangle 2D Construct
RF	Lattice Boltzmann Grid Refinement Factor
RMS	Root Mean Square
RWV	Rotating-Wall Vessel
SBB	Shifted Bounce-Back Boundary Treatment
SGA	Main Grid Area Covered by the Sub-Grid
SPH	Sphere 3D Construct
SQ	Square 2D Construct
UL	Upper Left Corner of the LDCF Geometry
UR	Upper Right Corner of the LDCF Geometry
VC++	Visual C++
VCF	Vertical Channel Flow Geometry
YU	Yu Linear Interpolation Boundary Treatment
c	Lattice Boltzmann Speed Constant
c_s	Speed of Sound
C	Dimensionless Drag Constant for Particle Sedimentation
δ_{ij}	Kronecker Delta
Δx	Lattice Spacing in the Lattice Boltzmann

Δt	Time Step in the Lattice Boltzmann
$\mathbf{e_i}$	Lattice Boltzmann Velocity Vector in *i*th Direction
$\mathbf{e_{ij}}$	Rate of Strain Tensor
E	Dimensionless Velocity Constant for Particle Sedimentation
f_i	Particle Distribution Function in *i*th Direction
$f_{i'}$	Particle Distribution Function in Direction opposite the *i*th Direction
f_i^c	Post-Collision Particle Distribution Function in *i*th Direction
f_i^{eq}	Equilibrium Particle Distribution Function in *i*th Direction
$\mathbf{F}$	Force Vector
I	Moment of Inertia Component for a Rigid Body
$\mathbf{I}$	Moment of Inertia Tensor
$\mathbf{L}$	Angular Momentum Vector
N	Lattice Boltzmann Grid Refinement Factor
M	Mass of a Rigid Body
Ma	Mach Number
μ	Dynamic Viscosity
ν	Y Component of Velocity Vector
Ω_i	Lattice Boltzmann Collision Operator
$\mathbf{\Omega_p}$	Angular Velocity Vector for a Particle
P	Pressure
$\mathbf{P}$	Linear Momentum Vector
ρ	Density
ρ_m	Mean Fluid Density

$\mathbf{R}$	Orientation Tensor
S	FLUENT Shear or Strain Rate
$\boldsymbol{\sigma}_{\mathbf{ij}}$	Stress Tensor
t	Time
$\mathbf{T}$	Torque Vector
τ	Lattice Boltzmann Relaxation Parameter
$\mathbf{u}$	Velocity Vector
$\mathbf{u}_w$	Wall Velocity Vector
$\underline{u}$	Velocity Magnitude
u	X Component of Velocity Vector
$\mathbf{U}_\mathbf{p}$	Velocity Vector for a Particle
υ	Kinematic Viscosity
w_i	Lattice Boltzmann EPDF Weighting Factor
$\mathbf{x}$	Position Vector
$\mathbf{Y}$	Matrix Describing State of a Rigid Body

Summary

Articular cartilage has a limited ability to heal due to its avascular, aneural, and alymphatic nature. Currently, there is a need for alternative therapies for diseases that affect articular cartilage such as osteoarthritis. Recently, it has been shown that tissue constructs, which resemble cartilage in structure and function, can be cultured *in vitro* in a cell-polymer bioreactor system. Bioreactors provide a three dimensional environment that promotes cell proliferation and matrix production. The primary objective of this study is to accurately simulate fluid mechanics using the lattice Boltzmann method for application to a cell-polymer bioreactor system.

Lattice Boltzmann (LB) is a flexible computation technique that will allow for the simulation of a moving construct under various bioreactor conditions. The method predicts macroscopic hydrodynamics by considering virtual particle interactions. Derived from the Lattice Gas Automata, lattice Boltzmann allows for mass transfer, complex geometries, and particle dynamics. A primary goal is to characterize the accuracy of the LB implementation and eventually the shear stresses felt by a tissue construct in this dynamic environment. This information is important since recent studies show that chondrocytic function may depend on the mechanical stimuli produced by fluid flow. Hence, shear stress may affect the final mechanical properties of tissue constructs. In this study, numerical simulations are done first in 2D and then extended to 3D to test the LB implementation. Simulations of the rotating wall vessel (RWV) bioreactor are then undertaken. The results are benchmarked against computations done with a commercial CFD package, FLUENT, and compared with analytic solutions and experimental data.

Chapter 1

Introduction

1.1 The RWV Bioreactor

In recent years, there has been a growing interest in the cultivation of cells in a three dimensional environment. Tools that facilitate the study of the adaptation of a cell have contributed to a deeper understanding of cellular function. The development of rotating-wall-vessel (RWV) bioreactors has allowed researchers to culture various cell types in a hospitable, three-dimensional environment. This environment promotes the co-location of cells, low shear stresses, mass transfer of nutrients and metabolic wastes, and three-dimensional growth (Spaulding et al. 1993). Since its development, several applications for this bioreactor environment have been explored including cancer research, drug efficacy, and infectious disease studies (Dooling 1998). Recent studies involving the tissue engineering of articular cartilage have resulted in successfully cultivated tissue constructs in this RWV bioreactor environment.

The interest in the tissue engineering of articular cartilage (AC) stems from the fact that AC is important in distributing loads in joints and has a limited ability to heal itself. This limited ability to heal is due to the characteristics of the main components of AC, chondrocytes (5% by volume) and extracellular matrix (ECM) mostly composed of collagen type II and proteoglycans. In damaged AC, mature chondrocytes cannot replicate or replace themselves and AC will replace its own ECM with fibrous tissue since its ECM has a slow turnover rate. This fibrous tissue is incapable of withstanding physical loading and eventually deteriorates, commonly seen in diseases such as osteoarthritis (Mow and Ratcliffe 1997). Hence, ECM is an important component in maintaining the natural mechanical properties of AC. Current treatments such as autografts, allografts, and total joint arthroplasty are hindered by a limited supply of fresh cartilage for transplant, potential for transmission of infectious disease, and poor adherence to bone substructure respectively. Thus, the therapeutic possibility of tissue engineered cartilage implants is attractive for treating damaged AC. Researchers have explored the possibility of culturing cartilage for transplant and repair of damaged AC. It has been shown that tissue constructs, which resemble natural cartilage in structure and function, can be grown *in vitro* in a cell-polymer bioreactor system (Freed et al 1998). In order to further understand the process by which AC is grown in the RWV bioreactor this study will attempt to characterize the fluid dynamics within the cell-polymer bioreactor system using numerical methods.

1.2 Problem Overview

The fluid dynamics of the RWV bioreactor are made complex by several factors including:

I. Geometry of the RWV bioreactor – The bioreactor consists of two concentric cylinders with culture medium filling the annular space. The inner and outer cylinders rotate at independent speeds, which are monitored and varied during the culture process.

II. Moving tissue construct – The AC tissue constructs are composed of a porous polymer scaffold onto which the chondrocytes are seeded. These constructs are allowed to float and move freely in the annular space of the RWV bioreactor.

III. Resolution of shear stresses – The shear stresses felt by a construct are of primary interest because it has been shown that cell metabolism is responsive to certain levels and types of shear stress (Malek & Izumo 1995). Resolving these stresses would provide insight into the adaptation process of cells within a construct.

It is necessary to address these issues if an accurate characterization and simulation of the fluid dynamics of the RWV bioreactor is to be accomplished. Research in numerical methods has shown the ability in resolving these types of issues.

1.3 Numerical Method

A computational model of the RWV bioreactor would require the ability to accurately reproduce the translating curved wall boundaries and the dynamics of a moving construct in the flow. In addition the model must allow the examination of refined velocities around the construct in order to obtain accurate shear stresses. Flexibility in the computational model is also a concern if additional issues such as the porosity of the construct and mass transfer characteristics of the flow are to be considered in the future. Currently, there are many numerical methods being used for computing multiphase fluid flows, whether fluid-fluid or solid-fluid interactions are under consideration. Various techniques employ Eulerian or Lagrangian approaches for tracking the evolution of interfaces (van Wachem 2003). However, with most of the numerical methods used for multiphase flows there are advantages and drawbacks. A review of methods (Prosperetti and Tryggvason 2003) reveals that many approaches are promising and will eventually contribute to significant advances in the field. One of these approaches is the lattice Boltzmann method. Recent developments in the lattice

Boltzmann method have made it a viable alternative to conventional Navier-Stokes solvers. The lattice Boltzmann method is a numerical scheme that is based on microscopic models and mesoscopic kinetic equations versus being based on discretizations of the macroscopic continuum equations (Chen and Doolen 1998). The primary advantages of the lattice Boltzmann method include ease in local explicit computations, flexibility in applications, and natural parallelism. In the past decade, work has been done with improved curved boundary treatments (Mei et al. 1999), multiphase flows, fluid-particle interactions (Aidun et al. 1998, Ladd & Verberg 2001), flow through porous materials (Koponen et al. 1998), and mass transfer flows. For these reasons, the lattice Boltzmann method was chosen as the numerical method for the computational model of the RWV bioreactor. The lattice Boltzmann method is a natural fit for this problem and holds enormous potential for providing insight into the construct growth process. This study will implement, innovate, and evaluate the performance of the lattice Boltzmann method for the RWV bioreactor problem.

1.4 Solution Methodology

This study will develop a computational model based on the lattice Boltzmann method that will accurately simulate the fluid dynamics of the RWV bioreactor. In the process of developing this model, certain of the aspects of the basic lattice Boltzmann method will be examined, implemented, evaluated, and in some cases innovated upon. Specifically, an interpolation boundary treatment will be proposed that handles curved, translating, and moving boundary types. Grid refinement and particle dynamics will also be implemented and evaluated for accuracy. To this end an excellent understanding is needed of the lattice Boltzmann method, boundary treatments, particle dynamics, and grid refinement. The model will initially be implemented and evaluated in two dimensions to verify its accuracy and then extended to three dimensions. Code for the computational model will be written in Visual C++ and MATLAB. This combination provides the flexibility of C++ object oriented programming with the built-in mathematical functionality of MATLAB. Verification of the accuracy of the model will be done through a series of numerical studies. The results of these studies will be compared with either numerical computations or experimental results. Verifying accuracy through comparisons with numerical computations will be done by comparing the lattice Boltzmann numerical results with results generated by a commercial CFD package, FLUENT. Comparisons of computational results to experimental particle image velocimetry data will be performed for validation purposes as well. This methodology will result in a thorough evaluation of the accuracy of the lattice Boltzmann method. The results of the study will not only aid the study of the RWV bioreactor problem, but also will aid the further development of the lattice Boltzmann method.

1.5 Organization of Thesis

This thesis is organized into six chapters. Chapter 1 is the overview and introduction. Chapter 2 is the review of previously published literature which provides the context for the current research. This chapter details the previous experimental work done with the RWV bioreactor and the basics of the lattice Boltzmann numerical method. Chapter 3 explains the current implementation of the lattice Boltzmann method. The current research brings together several aspects of the lattice Boltzmann method into one implementation which can be applied to the RWV bioreactor geometry. The following chapter contains an explanation of the numerical studies performed and the evaluation techniques used. Chapter 4 also presents the 2D and 3D results of the numerical studies. The discussion of the results is presented in Chapter 5. This chapter summarizes the key outcomes that verify and validate this implementation of the lattice Boltzmann method. Finally, Chapter 6 highlights the contributions of the current research and recommends areas for future research.

Chapter 2

Literature Review

The rotating-wall-vessel (RWV) bioreactor and the lattice Boltzmann method have been topics of interest for researchers for the past ten years. In particular, numerous studies involving the RWV bioreactor have been published in the 1990's that have examined different aspects of articular cartilage tissue engineering. In the past five years, research involving the lattice Boltzmann method has grown immensely, which is demonstrated by the recent papers published on a wide array of applications. In order to build a simulation model of the RWV bioreactor intelligently, an understanding of the bioreactor and the numerical method must be developed. This literature review will first address bioreactors, the work done with the RWV bioreactor, and the rising need for the characterization of the fluid dynamics of the bioreactors. Next, the review will address numerical methods and commercial solvers used to solve fluid dynamic problems, the background of the lattice Boltzmann method, and some lattice Boltzmann applications. Finally, the details of the method will be reviewed in order to understand its implementation of boundary treatments, particle dynamics, and grid refinement.

2.1. Bioreactors

Bioreactors are useful tools for scientists who wish to control the environments in which they culture living tissues. Several common types of bioreactors have been developed to grow viruses, bacteria, antibodies, or pharmaceuticals. However, a more recent application of bioreactors is tissue engineering. Research done with cell-polymer bioreactor systems has been promising in that tissue grown in these bioreactors exhibit traits of natural tissue found in the human body. Typically, cells are first seeded onto microcarrier beads or polymer scaffolds and then cultured into a larger tissue mass over time inside bioreactors. The long-term goal in this research is to develop methods and processes that will produce surgically implantable or clinically useful tissue that can function and integrate easily within the body. Consequently, designing bioreactors to that end will be an important area of research in the coming years.

2.1.1. Overview

A type of cell-polymer bioreactor system involves isolated cells that are seeded onto biodegradable polymer scaffolds and subsequently cultured in a bioreactor environment. In the case of the RWV bioreactor chondrocytes or cartilage cells have been used extensively. The polymer scaffolds provide the initial matrix for cells to adhere to until they produce their own extracellular matrix (ECM). For chondrocytes this ECM includes proteoglycans and type II collagen, which are critical components for proper tissue function. The polymer scaffolds are typically made of poly glycolic acid (PGA), poly l-lactic acid (PLLA), or poly lactic-co-glycolic acid (PLGA) formed in a mesh. As ECM production occurs the scaffold degrades and the tissue construct begins to resemble natural cartilage. The RWV bioreactor encourages ECM production and proliferation within the construct by creating a three-dimensional environment that promotes co-location and tissue differentiation, and induces low shear stresses and high mass transfer rates (Spaulding et al. 1993). These characteristics are achieved through the design of the RWV bioreactor, which consists of two concentric cylinders that rotate at various speeds with culture medium placed in the annular space. The inner cylinder serves as a filter for the exchange of gases while medium is replaced by perfusion. In some alternative rotating-wall-vessel bioreactor designs, the medium is exchanged by a batch method rather than perfusion. The cylinders slowly rotate at typical speeds between 10 to 70 rpm to provide a stable laminar flow field. Recent studies that use this RWV bioreactor design have successfully cultured chondrocytes isolated from bovine articular cartilage. Figure 2.1 illustrates a cross section of the RWV bioreactor with typical length and velocity scales.

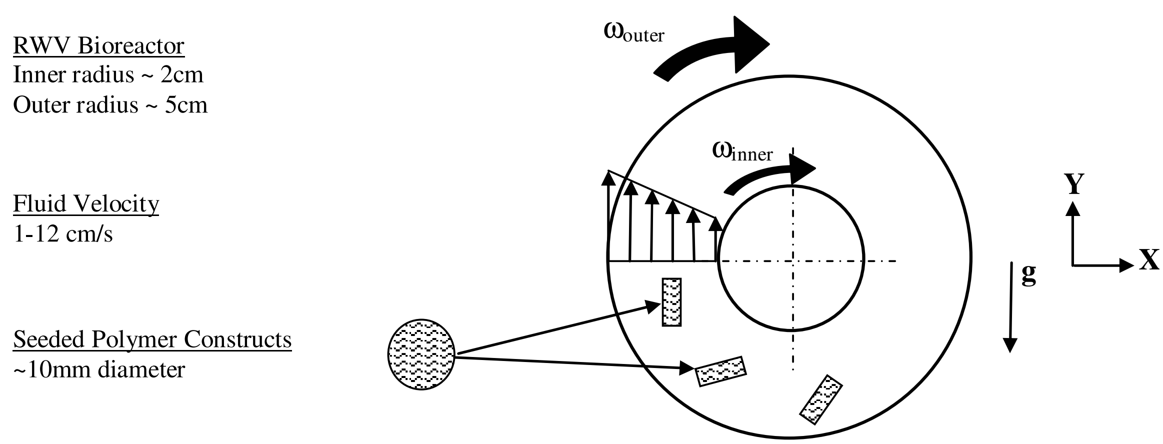


Figure 2.1 – The RWV Bioreactor

2.1.2. RWV Bioreactor Studies

Several studies over the past ten years have validated the potential of bioreactors in articular cartilage applications. It was demonstrated by Freed et al. (1993) that cartilage could be successfully grown on a polymer scaffold in a bioreactor. Freed et al. note that the potential advantages of bioreactors over static Petri dishes include (i) uniform mixing with control of mass transfer rates, (ii) regulation of shear stress, (iii) maintenance of constant pH, gas partial pressures, and nutrient levels, and (iv) the ability to develop process control strategies over the cultivation time. According to Freed, future studies would aim to relate chondrocyte growth and differentiation to the tissue culture environment and evaluate optimal conditions for bioreactor cultivation of clinically useful cartilage implants. Consequently, Freed and her colleagues have published studies that have explored these issues.

Freed et al. (1998) reported on experiments involving the RWV bioreactor in which several properties of articular cartilage tissue constructs grown over a forty days were examined. From the histological and biochemical analysis of the cartilage constructs, it was shown that they exhibited chondrocytic phenotype stability and that gylcosaminoglycan (GAG is an indicator of proteoglycan content) and collagen II wet weight fractions reached 68% and 33% of native cartilage values. While these values were promising and an improvement over tissue cultured in static Petri dishes, these constructs were still far from being clinically useful.

Obradovic et al. (1999) demonstrated that differences in cell metabolism and tissue structure resulted from controlling the concentration of metabolites inside the bioreactor. In these experiments with the RWV bioreactor, cartilage constructs were subjected to varying levels of gas and medium exchange and cultured for five weeks. The analysis showed that when aerobic conditions were present more cartilage ECM developed in constructs, while anaerobic conditions slowed ECM production. Anaerobic conditions were a result of either diffusion limitations in the supply of oxygen or low oxygen tension in the medium.

Ma & Langer (1999) examined the mechanical function of cartilage constructs grown in the RWV bioreactor. Their goal was to examine the effects of long term cultivation on the constructs and examine their mechanical properties using confined compression tests. They found that they improved the aggregate modulus of tissue engineered cartilage from 14% to 40% of the value in native cartilage by extending the cultivation time from 12 weeks to 25 weeks. Furthermore the compressive modulus and apparent permeability both reached levels of natural cartilage. The mechanical properties are important for articular cartilage since it typically lines the load bearing surfaces in major joints in the body and is subject to everyday wear.

Recently, Pei et al. (2002) reported on the effects of process parameters on the cultivation of cartilage constructs. These included scaffold material, scaffold structure, and culture system. The RWV bioreactor was shown to be more effective than static Petri dishes in initial cell seeding for three-day cartilage constructs and in overall construct properties for one-month constructs. A more uniform distribution of cells, higher numbers of cells, and better GAG and collagen content were seen in the bioreactor cartilage constructs. In similar reviews Vunjak-Novakovic et al. (2002) and Martin et al. (1999) report the viability of using the RWV bioreactor over other types of bioreactors like mixed flasks and static cultures. In both studies, the laminar nature of the hydrodynamics of the RWV bioreactor resulted in constructs with better properties in comparison to those constructs exposed to the harsher turbulence in mixed flasks or the calm of static flasks. Martin et al. reported that cartilage constructs cultivated for seven months had values close to those of native cartilage. Vunjak-Novakovic et al. reported that mechanical properties of cultivated constructs correlated well with explanted cartilage with respect to ECM content.

These studies demonstrate the potential of the RWV bioreactor environment for cultivating articular cartilage. Also the studies have shown that certain process parameters of the RWV bioreactor have a significant effect on the final cartilage construct properties. However, the full optimization of this environment will require analyzing the hydrodynamic environment involved and studying its effect on the cells. In particular, the shear stress felt by the construct is of interest because it has been shown by research that cells will respond differently under different types of shear stress.

2.1.3. Cell Response to Shear Stress

Although relatively little research has been done directly with the RWV bioreactor environment and how hydrodynamic forces and shear stresses affect constructs, the effects of shear stresses on cell function in general have been documented for several years. It has been shown that cells are morphologically and chemically sensitive to the types of stresses to which they are exposed. Dewey et al. (1981) exposed endothelial cells to controlled levels of shear stress using a cone and plate apparatus. Endothelial cells exhibited a change in shape and orientation in response to shear stress of 5-10 dynes/cm^2. In addition, endocytic activity and platelet interaction increased. In a similar study, Levesque & Nerem (1985) demonstrated that endothelial cells elongate and align themselves with the direction of flow within a parallel plate flow chamber when subjected to shear stresses of 10, 30, and 85 dynes/cm^2. It was also postulated that the response of the cells was influenced by contact with neighboring cells. Malek & Izumo

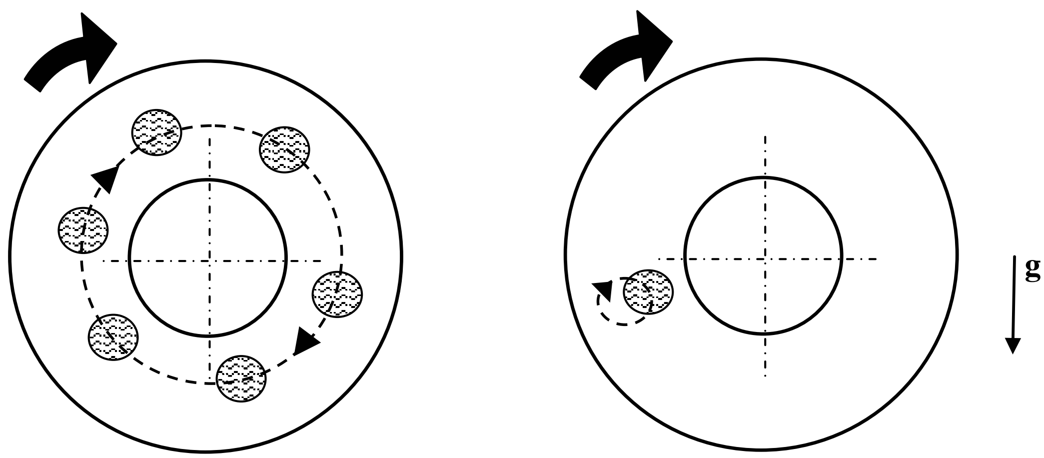
(1995) reported that endothelial cells appear to switch phenotype when subjected to low shear stresses of 15 dynes/cm^2. Under these conditions endothelial cells exhibited complex gene expression responses where more nitric oxide (NO is a vasodilator) and less endothelin-1 and platelet-derived growth factor B (ET-1 and PDGF-B are vasoconstrictors). Insight into these responses and their interaction with the cells around them in vivo will help explain the complex nature of vascular remodeling. Evidence presented in studies like these suggests that fluid shear stresses can play an important role in both physical and chemical cell response mechanisms.

Recently some studies examining the response of articular cartilage chondrocytes to mechanical and fluid shear stresses have been published. Smith et al. (1995) showed that chondrocytes, like endothelial cells, elongate and align themselves with the direction of flow. A chemical response was also observed as GAG synthesis and PGE_2 release were stimulated by the shear flow. In another study by Das et al. (1997), a flow-induced increase in GAG synthesis was shown to be dependent on several factors including NO synthesis and phospholipase C activation in bovine articular cartilage. In a review by Heath & Magari (1996), the influence of mechanical stresses on cartilage cultures in vitro was examined. They concluded that more research was needed to clearly define the response of cultured chondrocytes to mechanical stresses over long periods of time, but noted the promise of application of shear through fluid motion and the use of fluid convection to improve mass transfer. In a similar review of ex vivo synthesis of articular cartilage, Lebaron & Athanasiou (2000) postulated that focal adhesions allow the mechanical stimuli such as fluid flow to be transduced by cartilage cells and affect cellular function. Focal adhesions link the cytoskeleton to extracellular material as a conduit for chemical and mechanical signal transduction. Characterizing the stresses felt at the cellular level is important in further understanding transduction through these focal adhesions. This review noted the need for characterizing the fluid environments of bioreactors as the regulation of the biomechanical environment is a powerful modulator. Given that recent research support the fact that chondrocytes are sensitive to a fluid flow induced shear stress, it is reasonable to assume that the shear forces in the RWV bioreactor environment will have an effect on the metabolism and ultimately the mechanical properties of the cultivated cartilage.

2.1.4. Characterization of Fluid Dynamics in Bioreactors

The shear stresses felt by cartilage constructs inside the RWV bioreactor have been investigated but have not been thoroughly examined. However, there have been several studies that have looked at some of characteristics of the fluid dynamic environment in the RWV bioreactor. Wolf & Schwarz (1991) analyzed the gravity-induced motions of cultured tissues in the RWV bioreactor. They showed that a typical cultured tissue will attain 98% of its terminal velocity or Stokes velocity of 0.5 cm/s within 1 millisecond. In addition, the velocity of cultured tissue will be the vector summation of the Stokes velocity, the centrifugal-induced velocity, and the coriolis-induced velocity. One final conclusion stressed that complex secondary motions of particles are hard to characterize and test. While this study addressed the physical character of the culture environment, it did not specifically address the shear stresses on the cultured tissues.

Freed & Vunjak-Novakovic (1995) examined the relationship between hydrodynamic forces and chondrogenesis and illustrated typical dynamic behaviors of constructs in RWV bioreactors. Two regimes, the orbiting regime and the settling regime, are described based on construct behavior at varying wall speeds. In the orbiting regime, the construct will follow an orbital path around the center axis of the bioreactor. In the settling regime, the construct is in a state of continuous free-fall close to a stationary point. These two regimes are illustrated in Figure 2.2.



Orbiting Regime

Free-fall Regime

Figure 2.2 – Construct Behavior in the RWV Bioreactor

Figure 2.3 illustrates the force balance on a stationary construct. The drag force (F_d), buoyancy force (F_b), and gravitational force (F_g) are shown. In the settling regime, these forces are essentially balanced as the construct remains virtually stationary.

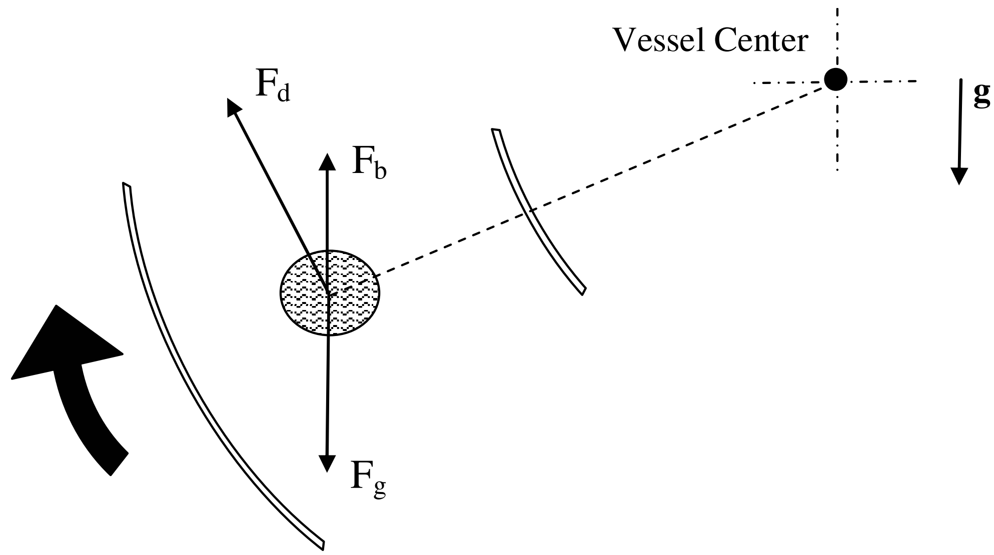


Figure 2.3 – Force Balance for a Construct in the RWV Bioreactor

This study estimated the average fluid shear stress on constructs to be 1.5 dynes/cm^2 on a one-week old construct. This average stress was calculated from the estimated drag force on the construct divided by the total surface area of the construct. The study concluded that the nature of the steady flow field created by a rotating wall bioreactor provides an excellent environment to examine the effects of hydrodynamic forces and stresses on tissue morphogenesis. However, Freed & Vunjak-Novakovic noted that further fluid dynamic and mass transfer studies would be necessary to fully understand the hydrodynamic environment.

An attempt to characterize the flow field in a RWV bioreactor experimentally was made by Brown (1998). In these experiments, particle image velocimetry (PIV) was used to study the flow around a fixed construct in a scaled model of a RWV bioreactor. The results of the PIV analysis and flow visualization studies suggest high velocity gradients near the surface of the constructs. These gradients made it difficult to accurately estimate the shear stresses. More recently, Sucosky et al. (2004) reported on experimental and computational results for spinner flask bioreactors. In this study, experimental PIV results were compared with computations done with FLUENT, a commercial CFD solver. Reasonable agreement between the PIV data and the computations was achieved despite the turbulent, complex nature of the flow field. In general, computational methods are a means to get meaningful estimates where experimental methods have encountered difficulty.

Recently, Begley & Kleis (2000) characterized the flow in a rotating-wall-perfusion-vessel (RWPV) bioreactor that uses a perfusion method to exchange culture medium. RWPV differs from the RWV bioreactor in that a thin solid disk, with a radius almost as large as the outer cylinder, is attached to and rotates with the inner cylinder at one end of the bioreactor. A numerical model was developed which solves the momentum and continuity equations in cylindrical coordinates for the flow. The model was used to examine the mean shear stress levels with the fluid under various operating conditions. Velocity results were compared with laser-Doppler velocimeter (LDV) experiments and showed good agreement. However, these results focused on the environment alone, without the effects of tissue microcarrier beads that would inhabit the flow.

Another recent study (Williams et al. 2002) models steady state conditions in a concentric cylinder bioreactor. The bioreactor used in this study was different from the RWV bioreactor in that the cylinders were vertically oriented such that gravity acts in the direction of the long axis and the constructs were fixed to the inner wall. FLUENT was used to calculate flow fields, shear stresses, and oxygen transport around nonporous stationary cartilage tissue constructs. The constructs themselves were fixed on the inner wall of the bioreactor space sufficiently apart. Results detailed a uniform shear stress distribution felt by the construct. In addition, an oxygen transport model was used to examine the effects of scaling with respect to oxygen depletion in the bioreactor.

While there have been several experimental and computational studies done characterizing the flow within bioreactors, there is still a need for more research in this area. The complex nature of a moving tissue construct inside the RWV bioreactor will be addressed by this study. A numerical model will be developed to help accurately resolve the shear stresses on a moving tissue construct. This model is based on the lattice Boltzmann method.

2.2. The Lattice Boltzmann Method

The lattice Boltzmann method is a fairly new computational method that has been the topic of several research papers published in the past ten years. Specifically, in the past five years the number of papers published that have implemented this method has grown considerably as the lattice Boltzmann method has become more popular. Some of the earlier works aimed at examining the accuracy, boundary treatments, and applications for the lattice Boltzmann method. More recent studies have reported results for its application to specific problems or situations. The lattice Boltzmann method has evolved as research has progressed, much in the same way other computational methods have evolved over time. While there is a tremendous amount of work done on specific aspects and applications of the lattice Boltzmann method, this literature review will attempt to cover some of the more basic elements of the lattice Boltzmann, particularly those that are applicable to the RWV bioreactor problem.

2.2.1. Numerical Methods and Commercial Solvers

There are numerous computational methods available to solve a wide range of fluid dynamics problems. In general, all methods are solving the continuity and momentum equations in addition to the energy equations in some cases. There are many approaches and variations to solving these equations including using direct numerical simulation (DNS) versus using averaged equations. For discretization of the equations one may apply the finite element method (FEM), finite difference method (FDM), or the finite volume method (FVM) to name a few. In the case of solving fluid problems involving particles in the flow one may take a Lagrangian versus an Eulerian approach. For simulating turbulent flows there are several methods that may be employed including large-eddy simulation (LES) or solving the Reynolds-averaged Navier-Stokes (RANS) equations. The preference for a particular method often depends on the problem under consideration, what computational code may available from previous work done within their research group, or what computational methods have freely available codes that can be easily adapted. Due to the difficulties in developing fully parallel solvers that can handle complex multiphase flow problems from scratch, Prosperetti & Tryggvason (2003) note that it is often beyond the scope of what a PhD student or small group of researchers can accomplish over the span of a few years. Alternatives to developing a code for solving fluid flow problems are the several commercial CFD packages available that will solve a more limited set of problems. These codes often trade robustness for accuracy and give limited access to the actual working code itself. However, the advantage of using a pre-packaged CFD solution is that the less time needs to be spent programming and configuring the code in order to compute solutions. Further advances in numerical methods solving multiphase flow have given researchers more options to only using these CFD pre-packed solutions.

There are several multiphase methods that have recently been developed that can track the deformation of an unsteady fluid interface separating phases. For a more thorough review the reader should see articles referenced by Prosperetti & Tryggvason (2003) and Van Wachem & Almstedt (2003). These methods include regular, fixed grid implementations like the volume of fluid (VOF) method using a level set method to improve accuracy at the interface (Sethian 2001) and front-tracking methods (Tryggvason et al. 2001). Other methods use boundary fitted grids for each phase (Takagi et al. 1997) or Lagrangian methods with grids that follow the fluid (Hu et al. 2001). The lattice Boltzmann method is just one of many methods that can simulate complex multiphase flow. Each method will exhibit advantages over other methods within a specific set of problems and therefore there is no clear universal choice.

Currently, there are several numerical methods that researchers have used to solve problems involving solid-fluid interactions where the lattice Boltzmann method has also been commonly applied. These methods have been specifically applied to problems involving solid particle dynamics in Newtonian and non-Newtonian fluids. Hu et al. (1992) and Feng et al. (1994) modified the Navier-Stokes POLYFLOW to solve the motion of a particle. This involves the calculation of force and torque on a particle, updating the motion of the particle using Newton's law, remeshing a new domain while incorporating the particles motion into boundary conditions, and solving by finite element. This method has evolved into the arbitrary Lagrangian-Eulerian (ALE) technique described by Hu et al. (2001). The updated technique is described as being based on a combined formulation of the fluid and particle momentum equations and a moving, unstructured, finite element mesh. In contrast, a distributed Lagrange multiplier-based fictitious domain method (DLM/FD) was proposed by Glowinski et al. (2001). This method does not compute forces and torques explicitly, but it is done implicated within the equations. This is done on a simple structured grid rather than a boundary fitted grid. The method is more akin to the front-tracking method mentioned previously where one set of equations is used for the whole domain. Lastly, a space-time finite element method has been proposed to solve moving boundary problems. These methods are a more general form of the ALE technique and have been use to solve sedimentation of a large number of particles (Johnson & Tezduyar 1997). While there may be advantages to using other numerical methods to solve the RWV bioreactor problem, for this study the lattice Boltzmann method was chosen for its potential ease of implementation, flexibility in application, and because there was no previous bias to any other methods.

2.2.2. Overview of the Lattice Boltzmann Method

The lattice Boltzmann (LB) method is a relatively new computation method that can accurately simulate macroscopic fluid behavior using microscopic models. This method differs from conventional methods that are based on solving the Navier-Stokes equations using the continuum assumption. For conventional computations, the momentum equations are solved with the continuity equation and the Poisson equation for pressure. This increases the number of operations in solving partial differential equations numerically. The LB method takes a different approach in that it is based on the movement of particles on a lattice. Although these particles follow simple rules at the microscopic level, their global behavior on the macroscopic scale produces the correct hydrodynamics. Its evolution has been steady ever since it emerged from lattice gas automata (LGA) and cellular automata (CA), the precursors to LB. LGA is a technique proposed by Frisch et al. (1986) to solve the incompressible Navier-Stokes equation. It is based on particles that inhabit a regular lattice and are moved from the present lattice site to a neighboring lattice site, a process known as streaming. When particles move to the same lattice site, a collision occurs and the particles are redistributed according to a set of collision rules. Conservation of particle number and particle momentum is guaranteed through these rules as well. At each time step, particles stream and collide at every lattice site. Symmetry in lattice geometry is important if the Navier-Stokes equation is to be recovered from the LGA (Frisch et al. 1986). The lattice used for LGA computations is usually triangular with hexagonal symmetry. It has been shown that the LGA produces good results; however there are several problems with the LGA that the LB method corrects. Two problems in the LGA include a non-Galilean invariance property due to density dependence on the convection coefficient and an unphysical and explicit pressure dependence on velocity (Satofuka & Nishioka 1999). Also, there is an inherent large amount of statistical noise in the LGA. The LBM corrects this by replacing the Boolean variables used to represent particle occupation of a lattice site with a particle distribution function represented by floating-point variables. This characterizes LB and sets it apart from the LGA.

There have been several excellent reviews on the LB method and its evolution from the LGA. Rothman & Zaleski (1994) gave an extensive review of lattice gas models and their applications to various problems such as multiphase flow. In this review, a detailed derivation of the lattice Boltzmann equation was given. This equation is the origin of both the LGA and LB. Benzi et al. (1992) focused more on the lattice Boltzmann method, a rigorous derivation of it, and some applications like 2D turbulence and flow in complex geometries. More recently, Chen & Doolen (1998) gave a thorough overview of the LB method and focused mainly on its potential applications such as flow through complex geometries, particles in flows, and multiphase flows. In a recent review, Yu et al. (2003) focused mainly on very recent advances in the LB method including single versus multiple relaxation-time models and multi-block methods. The interested readers are referred to these reviews for more in-depth study of LB.

2.2.3. Lattice Boltzmann Applications

There are several advantages to using the LB method that are not seen with conventional methods. Since the computations are localized the LB method lends itself nicely to parallelism. Recently, several studies have examined the advantages of programming LB on parallel systems (Ujita et al. 1998, Kandhai et al. 1998, Satofuka & Nishioka 1999). These studies have yielded positive results and encourage further investigations using the LB method on parallel systems. Another advantage in using the LB method is the ease of implementation of boundary conditions. The simplest boundary condition proposed is the bounce back rule whereby particle distributions reflect off solid boundaries and return to the lattice site from which they originated. Variations on this rule have yielded second-order accurate results. Other researchers have proposed slightly more complex boundary conditions (Noble et al. 1995, Zou & He 1997, Inamuro et al 1995) that have accurately modeled no-slip conditions on walls. Easily implemented boundary conditions allow for simulating fluid flows around complex geometries. This has led to studies examining flow through porous media (Koponen et al. 1998, Freed 1998, Inamuro et al. 1999). Lastly, the inherent flexibility of the LB method allows for the addition of several components to the simulation. Multicomponent flow (Grunau et al. 1993), various implementations simulating gravity (Buick & Greated 2000), heat transfer (Qian et al. 1993), and the addition of particle dynamics (Aidun et al. 1998, Qi 1999) are complex flow situations where the LB method has been implemented. These characteristics of LB can be advantageous in the simulation of the RWV cell-polymer bioreactor system.

2.2.4. Lattice Boltzmann Models

The LB method is similar to the LGA in that particles stream and collide on a symmetrical lattice for each time step. In addition the Navier-Stokes equations can be recovered from the lattice Boltzmann equation through a Chapman-Enskog expansion (Frisch et al. 1987). The lattice Boltzmann equation in its most general form is the following:

$$f_i(\mathbf{x} + \mathbf{e_i}\Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + \Omega_i(f(\mathbf{x}, t)) \qquad \mathrm{i} = 1, 2, ..., \mathrm{M} \qquad (2.1)$$

Equation 2.1 states that the streamed particle distribution function (PDF) f_i at the neighbor node at the next time step is the current particle distribution plus the collision operator Ω_i. The advection or streaming of a PDF occurs in the time Δt over a distance Δx which is the distance between lattice sites. Here $\mathbf{e_i}$ is the velocity vector in the *i*th direction, M is the total number of velocity vectors for the model, **x** is the position vector, and t is the current time. Again for LB, the f_i values are floating-point numbers rather than Boolean variables as in the LGA. The collision operator represents the rate of change of f_i resulting from collision (Chen and Doolen 1998). The collision operator can be further simplified into a linear collision term that relaxes the particle distribution function to an equilibrium state (Bhatnagar, Gross, and Krook 1954). This form the lattice Boltzmann equation is termed the LBGK equation and is as follows:

$$f_i(\mathbf{x}+\mathbf{e_i}\Delta t, t+\Delta t) = f_i(\mathbf{x},t) - \frac{f_i(\mathbf{x},t) - f_i^{eq}(\mathbf{x},t)}{\tau} \qquad (2.2)$$

The relaxation parameter is τ while f^{eq} represents the local equilibrium particle distribution function. The τ parameter is linked to the kinematic viscosity of the fluid, while f^{eq} is a function of the macroscopic velocity at each lattice site. Equation 2.3 is used to calculate kinematic viscosity.

$$\upsilon = (\tau - 1/2)c_s^2 \Delta t \qquad (2.3)$$

The kinematic viscosity υ, is a function of the relaxation parameter τ, the speed of sound c_s, and the time scale for the problem. The speed of sound is defined for the lattice Boltzmann model chosen. The viscosity is a positive parameter which requires that the relaxation parameter be larger than ½. The local equilibrium particle distribution function (EPDF) can be computed using Equation 2.4.

$$f_i^{eq}(\mathbf{x},t) = \rho\left[a + b(\mathbf{e_i}\cdot\mathbf{u}) + c(\mathbf{e_i}\cdot\mathbf{u})^2 + d\underline{u}^2\right] \qquad (2.4)$$

The density at the node is ρ, **u** represents the velocity vector, u̲ is the velocity magnitude, while constants that are specific to the lattice Boltzmann model chosen are given by *a*, *b*, *c*, and *d*. This formulation is valid for small velocities or small Mach numbers (Ma = u/c_s where c_s is the speed of sound). At each time step, collision occurs where by the particle distribution is relaxed toward equilibrium. These particle distributions are then streamed to their neighbor nodes. At this point certain considerations are handled like boundary treatments. Other processes that would also take place include particle dynamics. At the end of the time step the new local macroscopic properties can be calculated. The macroscopic properties of the flow, density ρ and momentum density ρ**u**, are defined in Equation 2.5 and Equation 2.6.

$$\rho = \sum_i^M f_i(\mathbf{x},t) \qquad (2.5)$$

$$\rho\mathbf{u} = \sum_i^M f_i(\mathbf{x},t)\mathbf{e_i} \qquad (2.6)$$

From these equations, the density and velocity can be calculated at each node. Another relevant equation that is computed at this time is the equation for the pressure. The pressure is calculated by the following:

$$P = c_s^2(\rho - \rho_m) \qquad (2.7)$$

The pressure at a node P depends on the speed of sound c_s, the current density at the node ρ, and the mean density of the fluid ρ_m. Again, the definition of c_s is dependent upon the model chosen. Upon completing the calculation for velocity, density, and pressure a new time step is begun and the process begins again. The general progression of the lattice Boltzmann method during one time step is illustrated below in Figure 2.4.

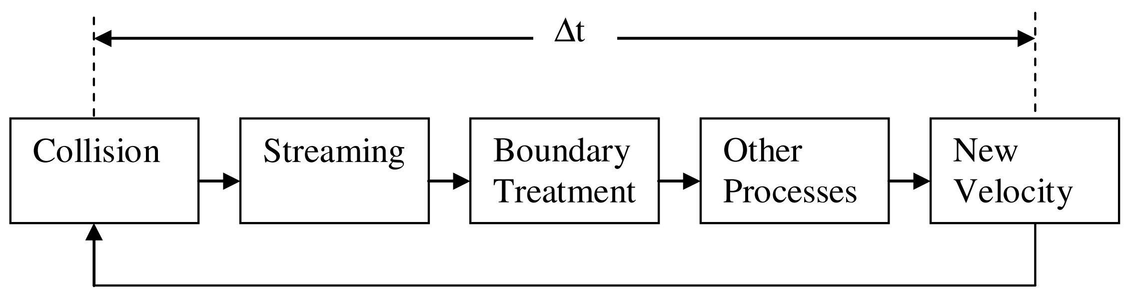


Figure 2.4 – A Typical Time Step in the Lattice Boltzmann Method

This process is repeated until the particle distribution functions are relaxed to an equilibrium state in the case of a steady flow problem. In the case of studying transient flow phenomena, the particle distribution functions and velocities at each node will not converge to an equilibrium value. Typically, for a steady problem a convergence criteria is needed and is often based on a relative error measurement.

In application, a lattice Boltzmann model must be chosen. The model can be based on a hexagonal, square, or cubic lattice arrangement. In addition for each model, the number of velocities needs to be specified. This will vary for a two dimensional implementation versus a three dimensional implementation. For the 2D model, a square or hexagonal model may be used. The hexagonal model had been a popular model used in the LGA, while the square model is the typical model used for LB. The square model is termed as the D2Q9 model. The name implies that the model is for two dimensions and at each lattice point there are nine velocities (M = 9) in which a particle can travel. One particular definition for these velocities is shown in Figure 2.5.

Velocity Magnitude I

$$\mathbf{e_i} = \left(\cos\left(\frac{\pi(i-1)}{4}\right), \sin\left(\frac{\pi(i-1)}{4}\right) \right) \frac{\Delta x}{\Delta t} \qquad \mathrm{i} = 1, 3, 5, 7$$

Velocity Magnitude II

$$\mathbf{e_i} = \sqrt{2}\left(\cos\left(\frac{\pi(i-1)}{4}\right), \sin\left(\frac{\pi(i-1)}{4}\right) \right) \frac{\Delta x}{\Delta t} \qquad \mathrm{i} = 2, 4, 6, 8$$

Velocity Magnitude 0

$$\mathbf{e_o} = 0$$

Figure 2.5 – Velocities for the D2Q9 Model

In Figure 2.5, the nine velocities are separated by their velocity magnitudes where group I velocities have a magnitude of one, group II velocities have a magnitude of √2, and the group 0 velocity has a magnitude of zero. A more visual interpretation of the velocity directions is illustrated below. Again, for each velocity direction *i* there is an associated PDF f_i at each node. When the particle streams with velocity $\mathbf{e_i}$, it takes with it the particle distribution function f_i to the neighboring node.

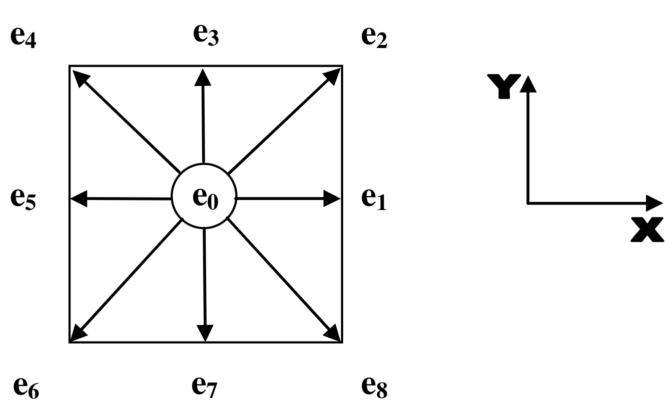


Figure 2.6 – Illustration of Velocities in the D2Q9 Model

The equilibrium particle distribution function f^{eq} that is involved in the collision step at the beginning of a time step can be defined for the D2Q9 model. Equation 2.8 is used to calculate the equilibrium function.

$$f_i^{eq}(\mathbf{x},t) = \rho w_i \left[1 + 3\frac{(\mathbf{e_i} \cdot \mathbf{u})}{c^2} + \frac{9}{2}\frac{(\mathbf{e_i} \cdot \mathbf{u})^2}{c^4} - \frac{3}{2}\frac{\underline{u}^2}{c^2} \right] \qquad (2.8)$$

The parameter w_i is a weighting factor specific for each velocity direction. In the case of the D2Q9, $w_0 = 4/9$, $w_I = 1/9$, and $w_{II} = 1/36$ where w_0 is the coefficient for the rest velocity, w_I is the coefficient for velocity directions with a magnitude of one (1, 3, 5, and 7 in this case), and w_{II} is the coefficient for velocity directions with a magnitude of $\sqrt{2}$ (2, 4, 6, and 8 in this case). The value of c is defined as $c = \Delta x/\Delta t$, which has a magnitude of one in this model. In addition, for this model, the speed of sound is defined as $c_s = c/\sqrt{3}$. This gives equations for the kinematic viscosity and the pressure as the following:

$$\upsilon = \frac{(2\tau - 1)}{6}\frac{\Delta x^2}{\Delta t} \qquad (2.9)$$

$$P = \frac{(\rho - \rho_m)}{3}\left(\frac{\Delta x}{\Delta t}\right)^2 \qquad (2.10)$$

The D2Q9 model is the most prevalent model used for a 2D formulation of the lattice Boltzmann method and has been well documented by the literature. Most if not all of the research done with the 2D lattice Boltzmann method is with the basic D2Q9 model.

For the 3D lattice Boltzmann models, a cubic lattice is often used. There are several different models that have been formulated. The main difference between these models is the chosen number of velocity directions. In general, the number of velocity directions will scale linearly with the number of operations required to run the model. Some examples of models reported in the literature include the D3Q15, D3Q19, and D3Q27 models (Qian et al. 1992). An example of the velocity directions of the D3Q19 model is illustrated in Figure 2.7. This model is similar to the D2Q9 model except that there are nineteen velocity directions instead of just nine. In the D3Q19 model there are also three velocity magnitudes as in the D2Q9 model.

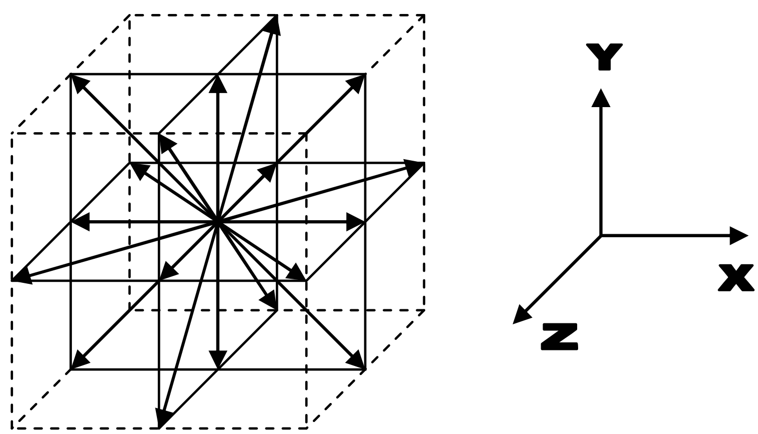


Figure 2.7 – Illustration of Velocities in the D3Q19 Model

The D3Q19 model has some similar characteristics to the D2Q9 model and may have some advantages over the D3Q15 model. As in the case of the D2Q9 model, the magnitude of the velocity a particle in the D3Q19 model can either have a value of zero, one or $\sqrt{2}$. Also in this model, particles will not travel to neighboring nodes located at the corners of the cube as illustrated in the figure. Another similarity is the use of the same equilibrium function as the D2Q9 model seen in Equation 2.8. The only difference being the weighting factors w. For the D3Q19 model, the weighting factors are $w_0 = 1/3$, $w_I = 1/18$, and $w_{II} = 1/36$. The parameters c_s and c are also the same for this model as the D2Q9 model. These similarities point to an easier implementation of the D3Q19 model, when coming from a D2Q9 implementation.

Some advantages are seen when examining the literature concerning the D3Q15 model and the D3Q19 model. A 14-directional lattice Boltzmann model developed by Chen et al. (1992) exhibited instabilities in the low-viscosity region. While this model used less memory than a 24-direction lattice model, the authors were not convinced that this model correctly handled incompressible Navier-Stokes flows. A few years later a good comparison of the D3Q19 model and the D3Q15 was reported in the literature by Kandhai et al. (1999). Kandhai et al. found that the D3Q19 model in practice did not show the artificial checkerboard invariants that the D3Q15 model displayed. The appearance of these invariants was most likely related to the lattice connectivity differences between the two models. However, Kandhai et al. did state that the D3Q15 model may be suitable for steady state hydrodynamic problems even with its shortcomings. In contrast the performance of the D3Q19 model was satisfactory. While other models like the D3Q27 model may perform as well as the D3Q19, it would require more memory and more operations during implementation. In addition, Mei et al. (2000) reported results for their boundary treatment using three models. They found that the D3Q15 model exhibited velocity oscillations and was prone to instability. Also, the D3Q27 model did not necessarily give more accurate results than the D3Q19 model. Thus, it is reasonable to conclude that a D3Q19 model is a good balance between the

D3Q15 and D3Q27 models. Considerations such as accuracy and ease of implementation need to be considered in choosing the right lattice Boltzmann model for a problem.

2.2.5. Boundary Treatments

Boundary conditions or boundary treatments for the lattice Boltzmann method have been an area of research interest for many years. The main problem researchers faced was to develop and implement a boundary treatment that was second-order accurate in space so as to match the accuracy of the lattice Boltzmann equation. Consequently, many boundary treatments were developed early on for the two dimensional D2Q9 lattice Boltzmann model. Some of these treatments were flexible and could be extended to a 3D lattice Boltzmann model, while some were not easily extended. A review of boundary treatments is necessary in understanding the actual implementation of the lattice Boltzmann method. The main issue a boundary treatment addresses in the lattice Boltzmann is how to fill PDF values in a node near a boundary if there is no existing node to stream it those PDF values. In other words, it must be shown how PDF values are accurately estimated at boundary nodes so that fluid conditions such as no-slip at the boundary are preserved.

Several boundary treatments have been proposed early on for 2D lattice Boltzmann models. The simplest type of treatment, which has roots in the Lattice Gas method, is the bounce-back scheme. In this treatment when a particle streams into a wall or lattice site representing something solid, it will bounce back from that site and return in the direction it originated. A particle will return to the lattice site or node it originated from in two time steps since the wall is located exactly one lattice distance away.

Although, it is simple to implement, this method has been shown to be first-order accurate and produced a slip velocity at the wall. A slight modification to this bounce-back method is the "shifted" or "halfway" bounce-back method. This method, which places an imaginary wall halfway between nodes, is second-order accurate. Equations 2.11 and 2.12 are used to calculate these two methods. Figure 2.8 illustrates the two methods.

$$f_{i'}\left(\mathbf{x_F}, t+2\Delta t\right) = f_i^c\left(\mathbf{x_F}, t\right) \qquad (2.11)$$

$$f_{i'}\left(\mathbf{x_F}, t+\Delta t\right) = f_i^c\left(\mathbf{x_F}, t\right) \qquad (2.12)$$

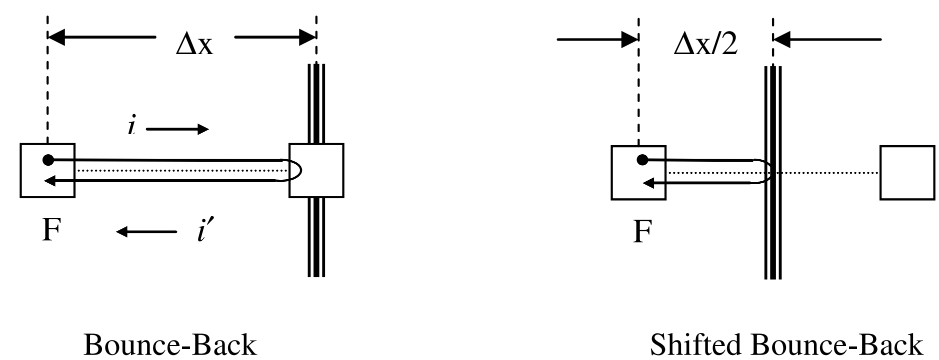


Figure 2.8 – Illustration of the Bounce-Back and Shifted Bounce-Back Treatments

Equation 2.11 says that the post-advection PDF value in the direction opposite of the wall *i´* at fluid node F will be equal to the post-collision PDF value in the direction of the wall *i* after two time steps. This is the bounce-back method. Equation 2.12 says that the post-advection PDF value in the direction opposite of the wall *i'* direction at fluid node F will be equal to the post-collision PDF value in the direction of the wall *i* after one time step. This is the shifted bounce-back method. While an improvement in accuracy is seen for the shifted bounce-back method, a slip velocity still existed at the wall. The other drawback of the bounce-back methods is that they don't lend themselves to accurately representing more complex boundaries. For example, with either method a curved surface must be represented with a stair step approximation of that surface. This effect can somewhat be mitigated by increasing the resolution of the lattice sites to get a smoother approximation of the surface, but this adds to computational time. The major advantage of the bounce-back methods is that they are simple, easy to implement, and cost little computational time.

Another set of boundary treatments proposed involves enforcing no-slip conditions at a boundary by setting particle values streaming into a flow at a boundary lattice site (Noble et al. 1995, Zou & He 1997, Inamuro et al. 1995). These boundary treatments, in general, are second order accurate and correctly produce a no-slip condition on the boundary. However, these boundary treatments require a node located at the boundary itself and are also restricted in representing complex boundaries. In addition, these methods are not easily extended into 3D. A concise summary of the details of the "hydrodynamic" boundary treatment by Noble et al. for the D2Q9 model is presented. The "non-equilibrium bounce-back" and the "non-slip" boundary treatments of Zou & He and Inamuro et al. are similar in style with the differences being noted.

The hydrodynamic boundary treatment calculation uses the conservation equations (Equations 2.5 and 2.6) for density and momentum density and adds an energy constraint. This yields four equations and four unknowns. To illustrate the calculation, consider a solid node that represents a wall below the flow regime in Figure 2.9.

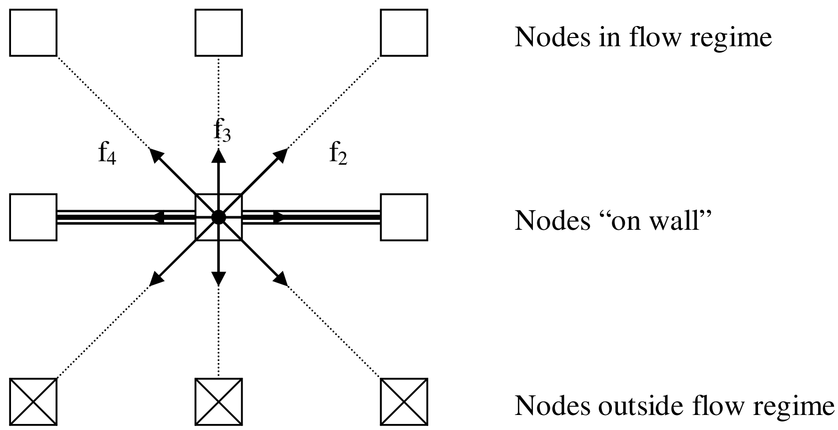


Figure 2.9 – Illustration for the Hydrodynamic Boundary Treatment

The particles distribution functions f_6, f_7, and f_8 stream to nodes not inside the flow regime. The links f_1 and f_5 stream to other solid nodes while links f_2, f_3, and f_4 stream to fluid nodes. The hydrodynamic equations for this situation are the following:

$$f_3 + f_2 + f_4 = \rho - (f_1 + f_5 + f_6 + f_7 + f_8 + f_0) \tag{2.13}$$
$$f_2 - f_4 = \rho u - (- f_5 - f_6 + f_8 + f_1) \tag{2.14}$$
$$f_3 + f_2 + f_4 = \rho v - (- f_6 - f_7 - f_8) \tag{2.15}$$
$$f_3 + 2f_2 + 2f_4 = 2\rho\varepsilon + \rho(u^2 + v^2) - (f_5 + 2f_6 + f_7 + 2f_8 + f_1) \tag{2.16}$$

The four unknowns for this set of equations are f_2, f_3, f_4, and ρ. Again f_2, f_3, and f_4 are unknown since there are no nodes inside the flow regime to stream values to these directions. Equation 2.13 is derived from the density conservation equation. Equation 2.14 and 2.15 are derived from the momentum density conservation equation (Equation 2.6), since that equation involves both the u and v velocity magnitudes of velocity **u**. Equation 2.16 is the energy constraint proposed by Noble et al. who prescribes the value of the parameter ε as the square of the speed of sound. The values of u and v are known at the wall so this set of equations can be solved directly for the four unknowns. Once the new values of f_2, f_3, and f_4 are known, these values are set for calculations of a new velocity and density, and are collided and streamed during the next time step.

The non-slip boundary treatment (Inamuro et al 1995) takes a slightly different approach. This treatment assumes that the unknown particle distribution function values are an equilibrium distribution function with a counter slip velocity magnitudes (u' and v'). This counter slip velocity ensures that the fluid velocity is equal to the wall velocity. If a solid node on the lower wall is considered again, these equilibrium equations are the following:

$$f_3 = \frac{1}{9}\rho'\left\{1 + 3v_w + \frac{9}{2}v_w^2 - \frac{3}{2}\left[(u_w + u')^2 + v_w^2\right]\right\} \tag{2.17}$$

$$f_2 = \frac{1}{36}\rho'\left\{1 + 3(u_w + v_w + u') + \frac{9}{2}(u_w + v_w + u')^2 - \frac{3}{2}\left[(u_w + u')^2 + v_w^2\right]\right\} \tag{2.18}$$

$$f_4 = \frac{1}{36}\rho'\left\{1 + 3(-u_w + v_w - u') + \frac{9}{2}(-u_w + v_w - u')^2 - \frac{3}{2}\left[(u_w + u')^2 + v_w^2\right]\right\} \tag{2.19}$$

These equations are coupled with Equations 2.13, 2.14, and 2.15 to form a set of six equations and six unknowns. The unknowns are f_2, f_3, f_4, ρ_w, ρ', and u' where ρ_w replaces ρ in Equations 2.13, 2.14, and 2.15, and $\rho'$and u' are additional unknown parameters from the equilibrium Equations 2.17, 2.18, and 2.19. The velocity magnitudes at the wall, u_w and v_w are also known. In order to solve this set of equations, ρ_w is solved for first from Equations 2.13 and 2.15. Next, ρ' is solved by eliminating f_2, f_3, and f_4 from Equations 2.17, 2.18, 2.19, and 2.15 and noting that $u_w + u'$ equals zero. Then u' is solved for using Equations 2.14, 2.18, and 2.19 by eliminating f_2 and f_4. Thus, knowing these parameters u' and ρ', the unknown PDF values f_2, f_3, and f_4 can be calculated using the equilibrium equations. The equations for ρ_w, ρ', and u' are the following:

$$\rho_w = \frac{1}{1-v}\{f_0 + f_1 + f_5 + 2(f_6 + f_7 + f_8)\} \qquad (2.20)$$

$$\rho' = 6\frac{(\rho_w v_w + f_6 + f_7 + f_8)}{(1 + 3v_w + 3v_w^2)} \qquad (2.21)$$

$$u' = \frac{1}{1+3v}\left\{\frac{6}{\rho'}[\rho_w u_w - (f_1 + f_8 - f_5 - f_6)] - u_w - 3u_w v_w\right\} \qquad (2.22)$$

While the non-slip boundary treatment is very complex, the non-equilibrium bounce-back treatment proposed by Zou & He is more simplified. It assumes that bounce-back for the non-equilibrium part of the particle distribution normal to the wall applies. This gives a fourth equation to solve for four unknowns. Again, in the case of a lower wall node these unknowns are f_2, f_3, f_4, and ρ. The first three equations are identical to Equations 2.13, 2.14, and 2.15, and the fourth equation is the following:

$$f_3 - f_3^{eq} = f_7 - f_7^{eq} \Rightarrow f_3 = f_7 + \left(f_3^{eq} - f_7^{eq}\right) \qquad (2.23)$$

Using the equilibrium equations for f_3 and f_7, f_3 can be solved for directly. The density can be solved in the same way the non-slip treatment solves for ρ using Equations 2.13 and 2.15. The result for ρ is the same as Equation 2.20. Finally, using Equations 2.14 and 2.15, expressions for f_2 and f_4 can be derived that include velocity magnitudes at the wall u and v. The final equations for f_2, f_3, and f_4 are as follows:

$$f_3 = f_7 + \frac{2}{3}\rho v \qquad (2.24)$$

$$f_2 = f_6 - \frac{1}{2}(f_1 - f_5) + \frac{1}{2}\rho u + \frac{1}{6}\rho v \qquad (2.25)$$

$$f_4 = f_8 + \frac{1}{2}(f_1 - f_5) - \frac{1}{2}\rho u + \frac{1}{6}\rho v \qquad (2.26)$$

These three boundary treatments have some unique characteristics along with some disadvantages. They all assume that the velocity at the wall is known and that density is unknown. It turns out that for the hydrodynamic and non-equilibrium bounce-back treatments, if you know the density and one velocity component, you can solve for the other velocity component. This flexibility allows for pressure boundary conditions to be applied at an inlet or outlet. Thus the hydrodynamic and equilibrium boundary treatments can be used for inlet and outlet conditions in the D2Q9 model. The disadvantages of these treatments include the fact that curved surfaces are problematic because the wall must be located one lattice distance away and that an extension of these treatments into 3D would require the solution of more equations and more unknowns. Thus, boundary treatments that could represent a curved surface accurately and be easily extended into 3D still needed to be developed.

In order to address these deficiencies, boundary treatments by Filippova & Hänel (1998a), Mei, Luo, and Shyy (1999), and the continuous bounce-back treatment by Verberg & Ladd (2000) emerged. These treatments accounted for a boundary at various distances between nodes. The Mei, Luo, Shyy (MLS) boundary treatment was proposed as an improvement to the Filippova & Hänel (FH) boundary treatment. Mei et al. (1999) reported that the MLS boundary treatment (i) preserves the geometry of interest without having to use a stair step approximation, (ii) generally results in solutions of second-order accuracy for the velocity in space and in time in some cases, and (iii) gives comparable or better results for the flow field over the shifted bounce-back scheme. In a follow up study Mei et al. (2000) reported results for the MLS boundary treatment in 3D flows. They reported that the flow through a square duct and circular pipe showed second-order accuracy and that flow over a sphere showed good self-consistency of the solution. A brief summary of the MLS boundary treatment equations and an illustration in Figure 2.10 are presented.

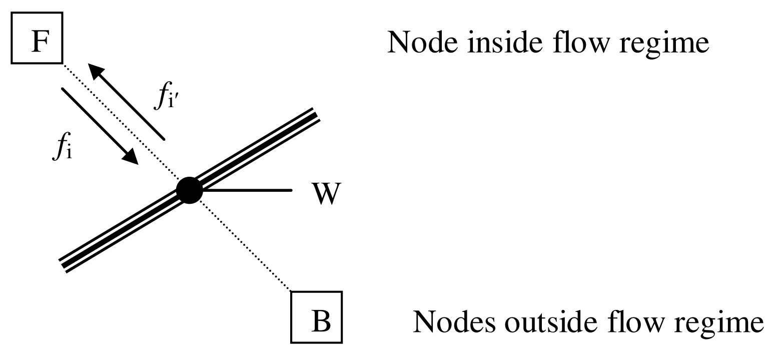


Figure 2.10 – Illustration for the MLS Boundary Treatment

$$f_{i'}^{c}(\mathbf{x_B},t) = (1-\chi) f_i^{c}(\mathbf{x_F},t) + \chi f_i^{*}(\mathbf{x_B},t) + 2 w_i \rho \frac{3}{c^2}(\mathbf{e_{i'}} \cdot \mathbf{u_W}) \tag{2.27}$$

$$f_i^{*}(\mathbf{x_B},t) = w_i \rho \left[1 + \frac{3}{c^2}(\mathbf{e_i} \cdot \mathbf{u_{BF}}) + \frac{9}{2c^4}(\mathbf{e_i} \cdot \mathbf{u_F})^2 - \frac{3}{2c^2} \underline{u}_F^2 \right] \tag{2.28}$$

$$\mathbf{u_{BF}} = \frac{(\Delta - 1)\mathbf{u_F}}{\Delta} + \frac{\mathbf{u_W}}{\Delta}, \qquad \chi = \frac{(2\Delta - 1)}{\tau}, \Delta \geq \frac{1}{2} \tag{2.29}$$

$$\mathbf{u_{BF}} = \mathbf{u_F}(\mathbf{x_F} + \mathbf{e_i}\Delta t, t), \qquad \chi = \frac{(2\Delta - 1)}{(\tau - 2)}, \Delta < \frac{1}{2} \tag{2.30}$$

$$\Delta = \frac{|\mathbf{x_F} - \mathbf{x_W}|}{|\mathbf{x_F} - \mathbf{x_B}|} \tag{2.31}$$

The MLS boundary treatment is based on an interpolation for f^c, the post collision PDF value that will stream to node F from node B in the i' direction. The value of f^c depends on χ a weighting factor, Δ the normalized distance of the node to the wall, and f^* an equilibrium distribution function that depends on $\mathbf{u_{BF}}$ a chosen velocity value. The choice for $\mathbf{u_{BF}}$ for $\Delta \geq ½$ is different for the MLS boundary treatment than the FH boundary treatment. The last term which includes the density ρ and the equilibrium weighting factor w_i, is a momentum exchange term for a moving wall that is seen in several boundary treatments. This term will be discussed more in the particle dynamics portion of the literature review. Some features to note include that the treatment has a different equation for $\Delta \geq ½$ and $\Delta < ½$, the choice of $\mathbf{u_{BF}}$ is not unique, and that the interpolation is linear in nature. These features distinguish its implementation from other treatments.

The continuous bounce-back method (CBB) by Verberg & Ladd (2000) introduced a continuous parameter α that represents the fluid volume fraction at each node and has the advantages of not requiring a surface normal and small additional computational overhead. The treatment is based on the link bounce-back method, or particles bouncing back someplace along the link between nodes. The rules for governing the PDF values are complicated to implement for different geometries like inclined boundaries, but a sample calculation for a simple case is presented. Consider the following diagram of a fluid node streaming a particle to a node that is partially filled with fluid.

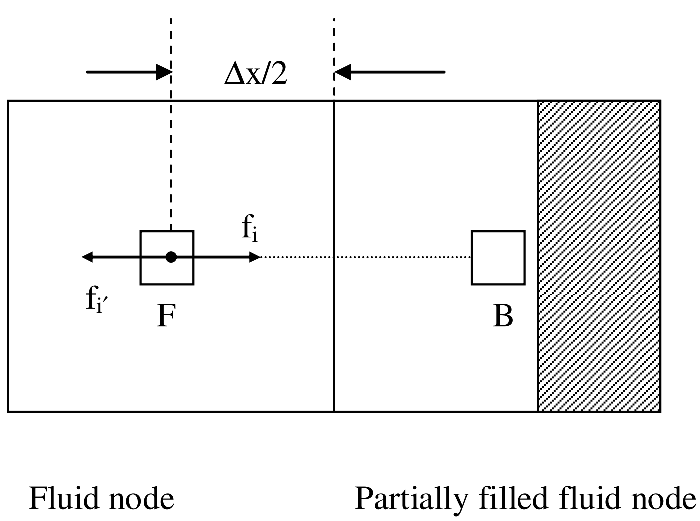


Figure 2.11 – Illustration for the CBB boundary treatment

$$f_{i'}\left(\mathbf{x_F}, t+\Delta t\right) = f_i^{\,c}\left(\mathbf{x_F}, t\right) \tag{2.32}$$

$$f_{i'}\left(\mathbf{x_B}, t+\Delta t\right) = f_i^{\,c}\left(\mathbf{x_F}, t\right) \tag{2.33}$$

$$f_i\left(\mathbf{x_B}, t+\Delta t\right) = \left(1-2\alpha_B\right) f_i^{\,c}\left(\mathbf{x_F}, t\right) + \alpha_B\left[f_{i'}^{\,c}\left(\mathbf{x_F}, t\right) + f_{i'}^{\,c}\left(\mathbf{x_B}, t\right)\right] \tag{2.34}$$

These equations are valid for $\alpha_B < 0.5$. If α_b is greater than 0.5 then Equations 2.33 and 2.34 must be modified. The treatment would reduce to the shifted bounce-back scheme for $\alpha_B = 0.5$. The computations are different and more complex for an inclined boundary. However, even though the computations may be complex in some cases, Verberg & Ladd report that treatment can reduce computation time and memory used through a reduction in resolution requirements. In a more recent publication, Verberg & Ladd (2001) report in detail the accuracy and stability of the continuous bounce-back boundary treatment. For the cases of 2D flow in inclined channels and flow throw a periodic array of disks or spheres, results showed that the hydrodynamic boundary is displaced from the physical boundary. However, the treatment has shown second-order and has been applied to porous geometries and particle suspensions with success.

Another set of boundary treatments that have appeared use simple extrapolation or interpolation rules to determine unknown PDF values. An extrapolation method presented by Chen et al. (1996) extrapolates particle values that would stream into the flow at a boundary. This boundary could be a solid boundary or a pressure or velocity inlet or outlet boundary. The primary advantage of the treatment is that it is simple to implement. Consider the following illustration in Figure 2.12.

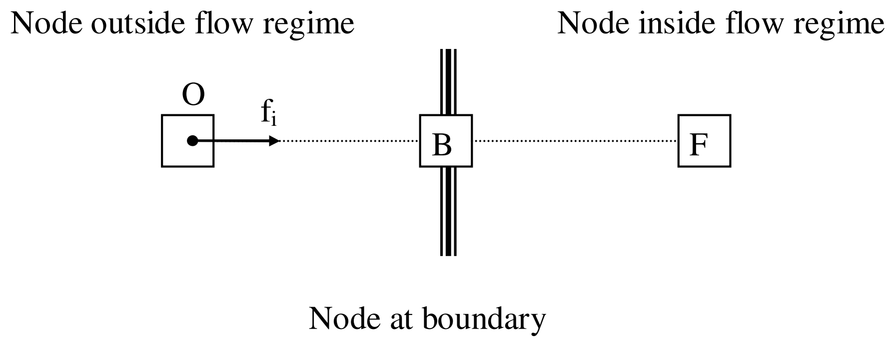


Figure 2.12 – Illustration for the Chen Extrapolation Boundary Treatment

$$f_i\ (\mathbf{x_b}, t + \Delta t) = f_i\ (\mathbf{x_o}, t) = 2 f_i^c (\mathbf{x_b}, t) - f_i^c (\mathbf{x_f}, t) \qquad (2.35)$$

The equation states that unknown post-advection PDF value at node B can be solved using a simple finite-difference formula involving the post-collision PDF values at the boundary node and the next node inside the flow regime. This method assumes that there is an imaginary node O outside the flow regime at a distance of Δx away. This method can be useful for pressure inlets or free outlets, but for solid boundaries the method becomes a bit more complicated. As reported by Chen et al. (1996), if the wall is placed in between the boundary node and the node inside the flow regime, then a velocity and density can be assigned to the boundary node and a PDF value calculated. Equation 2.35 then can be used to calculate a value for the node outside the flow regime. Chen et al. reports that the method displays second-order error convergence and is flexible enough to handle more than a non-slip condition at the boundary. Disadvantages include that extrapolation often has more error associated with it than interpolation, and that the method is problematic when two surfaces are close together because there are limited values available for extrapolation.

In general, interpolation rather than extrapolation can provide better results because the estimated value lies within the range of known values. An interpolation method has been presented by Bouzidi et al. (2001) that uses linear and quadratic interpolations in combination with the bounce-back method to calculate PDF values. The method is simple to implement can account for boundaries placed at any distance between nodes. Consider Figure 2.13 which illustrates two cases where the wall is less than one-half Δx and greater than one-half Δx.

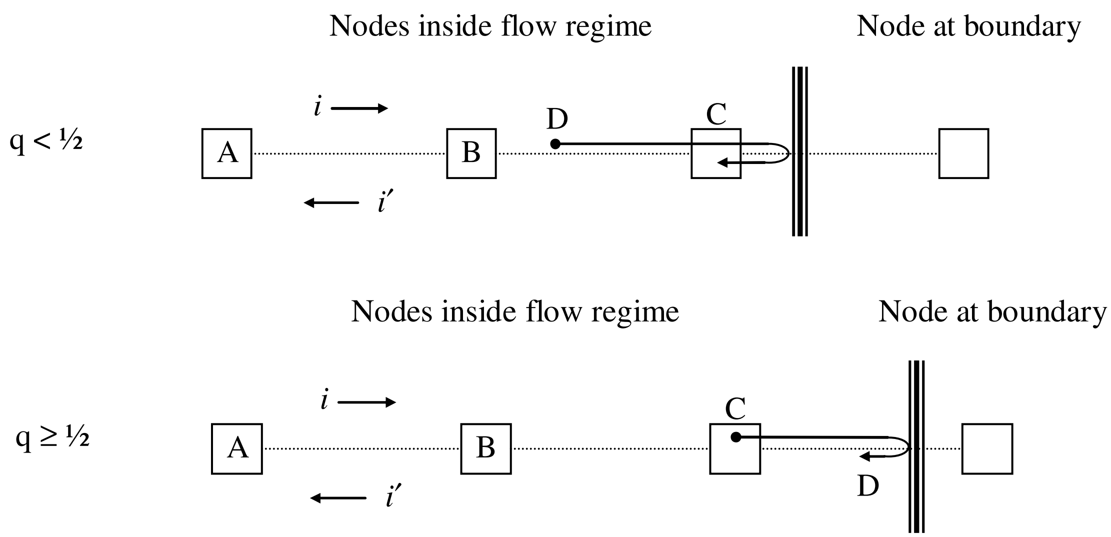


Figure 2.13 – Illustration for the Bouzidi Interpolation Boundary Treatment

$$f_{i'}(\mathbf{x_C}, t+\Delta t) = 2q \cdot f_i^c(\mathbf{x_C}, t) + (1-2q) \cdot f_i^c(\mathbf{x_B}, t) \tag{2.36}$$

$$f_{i'}(\mathbf{x_C}, t+\Delta t) = q(2q+1) \cdot f_i^c(\mathbf{x_C}, t) + (1+2q)(1-2q) \cdot f_i^c(\mathbf{x_B}, t) - q(1-2q) \cdot f_i^c(\mathbf{x_A}, t) \tag{2.37}$$

$$f_{i'}(\mathbf{x_C}, t+\Delta t) = \frac{1}{2q} \cdot f_i^c(\mathbf{x_C}, t) + \frac{(2q-1)}{2q} \cdot f_{i'}^c(\mathbf{x_C}, t) \tag{2.38}$$

$$f_{i'}(\mathbf{x_C}, t+\Delta t) = \frac{1}{q(2q+1)} \cdot f_i^c(\mathbf{x_C}, t) + \frac{(2q-1)}{q} \cdot f_{i'}^c(\mathbf{x_C}, t) + \frac{(1-2q)}{(1+2q)} \cdot f_{i'}^c(\mathbf{x_B}, t) \tag{2.39}$$

In the illustration, q is the normalized distance from the node (C in this case) to the wall. Note that q is normalized by the lattice spacing Δx. Equations 2.36 and 2.37 are the linear and quadratic interpolation for the post-advection PDF value at node C in the direction opposite of the wall (i') for $q < ½$. These formulas use the post-collision PDF values at A, B, and C in order to find the PDF value at D that will end up streaming to C after bounce-back at the wall. When $q \geq ½$, the post collision PDF values at B and C (one of which ends up at D after streaming) are used to interpolate the PDF value at C in the direction opposite of the wall. This is shown in the linear and quadratic interpolations in Equations 2.38 and 2.39. Bouzidi et al. reported that the boundary treatment could handle curved surfaces and moving boundaries by accounting for momentum exchange between boundaries and particles. Results for circular Couette flow and 2D steady flow past a periodic array of circular cylinders showed good agreement with solvers of the Navier-Stokes equations. The advantages of this method are that it is an intuitive concept as it is an extension of the bounce-back method and that it is easy to implement and extend to 3D. A disadvantage is that the boundary treatment is not the same for wall positions where $q < ½$ and $q \geq ½$, which is seen in previous boundary treatments.

More recent extensions of the Bouzidi boundary treatment have surfaced and have been reported in the literature. Yu et al. (2003) has presented a unified boundary treatment that simplifies the Bouzidi boundary treatment into one rule for all wall distances. Yu proposed first finding the post collision PDF value of a particle that would end up at the wall at the end of a time step. Then an equation could be derived that could always be used for interpolation, either linear or quadratic. Consider the following illustration in Figure 2.14.

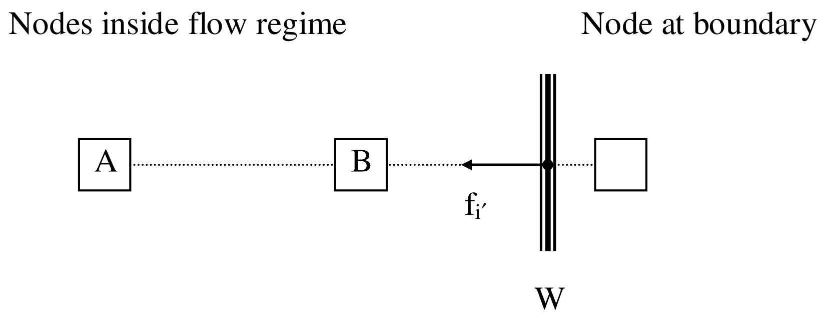


Figure 2.14 – Illustration for the Yu Interpolation Boundary Treatment

$$f_{i'}(\mathbf{x_W}, t+\Delta t) = f_i^c(\mathbf{x_A}, t) + q\left[f_i^c(\mathbf{x_B}, t) - f_i^c(\mathbf{x_A}, t)\right] \tag{2.40}$$

$$f_{i'}(\mathbf{x_B}, t+\Delta t) = f_{i'}(\mathbf{x_W}, t+\Delta t) + \frac{q}{1+q}\left[f_{i'}(\mathbf{x_A}, t+\Delta t) - f_{i'}(\mathbf{x_W}, t+\Delta t)\right] \tag{2.41}$$

The post-advection PDF value at x_w is linearly interpolated from post collision PDF values in Equation 2.40. The value q is again the normalized distance from the node (B in this case) to the wall. This wall value is then used in the unified formula, Equation 2.41, to linearly interpolate the post-advection PDF value at node B in the direction opposite the wall. This same process can be done using quadratic interpolations instead of linear interpolations to improve accuracy. Yu et al. (2003) in a recent review reports the results of simulations using this unified boundary treatment compared to the FH boundary treatment and the Bouzidi boundary treatment. The reported results showed that the drag coefficient on a cylinder and flat plate compared favorably to previous reported results for the unified boundary treatment compared to the FH and Bouzidi boundary treatment. The unified boundary treatment presented by Yu et al. has the same advantages as the Bouzidi treatment and also eliminates the separate calculations for different wall distances.

This study will present a boundary treatment that is an extension of the interpolation methods presented here. The new method can account for complex geometries, incorporates more PDF information from the surrounding nodes in order to increase accuracy, and can be used with moving boundaries for facilitation of particle dynamics.

2.2.6. Particle Dynamics

Simulations incorporating particle dynamics or moving objects using the lattice Boltzmann method have been presented in literature. For each simulation incorporating particle dynamics there are several key questions that need to be answered with regard to the lattice Boltzmann method. The first question to answer is how force is evaluated on a surface. The second question to address is how fluid nodes are treated if at some point in the simulation they become a solid node. The last question is how forces are evaluated when the distance between surfaces is less than one lattice spacing. These issues have been addressed and debated in the literature by several researchers over the past decade.

Ladd (1994a, 1994b) presented a comprehensive study on numerical simulations of particulate suspensions. In this study, Ladd laid down a theoretical foundation for a discretized Boltzmann model and reported on several numerical simulations. In a later study, Ladd & Verberg (2001) detailed the implementation of the lattice Boltzmann model with particle-fluid suspensions. In general, the method by which Ladd & Verberg evaluate forces on a surface depends on a momentum-exchange method. The interaction between particle momentum and a moving boundary located halfway between two nodes is illustrated below. The particles bounce-back from the boundary with altered momentum.

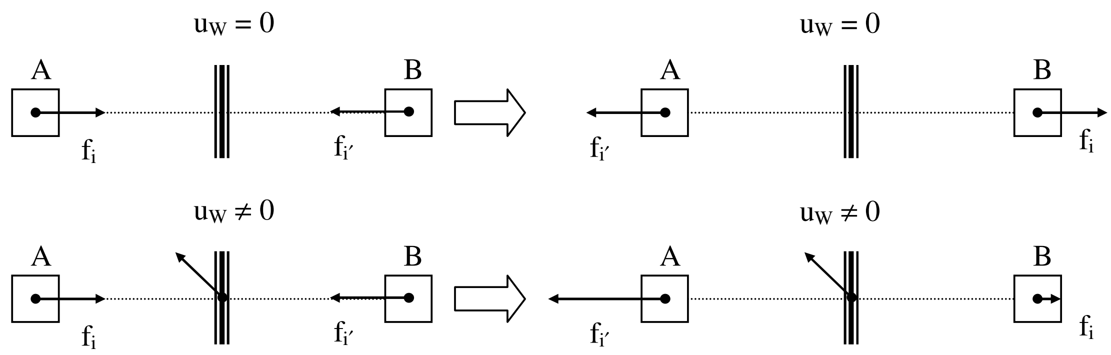


Figure 2.15 – Illustration of Momentum Exchange at a Moving Wall

$$f_{i'}\left(\mathbf{x_A}, t+\Delta t\right) = f_i^c\left(\mathbf{x_A}, t\right) - \frac{2 w_i \rho\left(\mathbf{u_W} \cdot \mathbf{e_i}\right)}{c_s^2} \qquad (2.42)$$

$$f_i\left(\mathbf{x_B}, t+\Delta t\right) = f_{i'}^c\left(\mathbf{x_B}, t\right) + \frac{2 w_i \rho\left(\mathbf{u_W} \cdot \mathbf{e_i}\right)}{c_s^2} \qquad (2.43)$$

Depending on which direction (i) is taken as reference with respect to the wall velocity ($\mathbf{u_W}$), momentum is either added or taken away from the particle by adjusting the post-advection PDF value. In the case of node A, momentum is added to the particle by adding to the bounce-back PDF value in Equation 2.42 because the dot product of the wall velocity and reference direction (i) is a negative value. The opposite is true in Equation 2.43 for node B. This idea of momentum exchange is the basis of the evaluation of force on a surface using momentum exchange. From these equations, the force exerted on the boundary can be calculated.

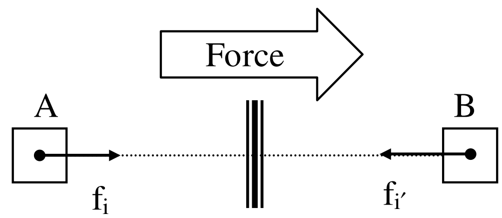


Figure 2.16 – Illustration of the Momentum-Exchange Method for Force Evaluation (fluid exists on both sides of wall)

$$\mathbf{F}(\mathbf{x_W}, t+\frac{1}{2}\Delta t) = 2\frac{\Delta x^3}{\Delta t}\left[f_i^c(\mathbf{x_A}, t) - f_{i'}^c(\mathbf{x_B}, t) - \frac{2w_i\rho(\mathbf{u_W}\cdot\mathbf{e_i})}{c_s^2}\right]\mathbf{e_i} \tag{2.44}$$

The force on the boundary **F** is calculated from the post-collision PDF values at A and B, and from the momentum exchange term related to the velocity of the wall $\mathbf{u_W}$. The direction of **F** is determined by the reference velocity $\mathbf{e_i}$ chosen in the calculation. In other words, the force acts along the link between the two nodes and in the chosen reference direction. In order to get the total force and torque on a solid moving particle immersed in fluid, a summation of the forces is done around the boundary of a particle.

$$\mathbf{F_{Total}}\left(t+\frac{1}{2}\Delta t\right) = \sum \mathbf{F}(\mathbf{x_W}, t+\frac{1}{2}\Delta t) \tag{2.45}$$

$$\mathbf{T_{Total}}\left(t+\frac{1}{2}\Delta t\right) = \sum (\mathbf{x_W} - \mathbf{x_{CM}})\times\mathbf{F}(\mathbf{x_W}, t+\frac{1}{2}\Delta t) \tag{2.46}$$

Total force $\mathbf{F_{Total}}$ is the summation of all the forces at the walls around the boundary of the particle. The force **F** at the wall of the particle is crossed with the position of the wall with respect to the center of mass with the particle $\mathbf{x_{CM}}$ in order to find the torque contribution **T** at that point. Total torque $\mathbf{T_{Total}}$ over the particle in Equation 2.46 is the summation of the torque contributions around the boundary. In the case of a moving particle, the velocity of the wall at a particular point can be found by the following equation.

$$\mathbf{u}_{\mathbf{W}} = \mathbf{U}_{\mathbf{p}} + \mathbf{\Omega}_{\mathbf{p}} \times \left(\mathbf{x}_{\mathbf{W}} - \mathbf{x}_{\mathbf{CM}}\right) \qquad (2.47)$$

The velocity at the wall $\mathbf{u_W}$ is a function of the particle velocity $\mathbf{U_p}$, the angular velocity of the particle $\mathbf{\Omega_p}$, and the position of the wall with respect to the center of mass of the particle $\mathbf{x_{CM}}$. Again, this velocity at the wall is needed to compute the contributing force at the wall on the particle with the momentum exchange method. The total force and torque are averaged over t + ½Δt and t – ½Δt in order to find the total force and torque at time t. Once this is known, the particle motion is then determined by solving the equations of motion for translation and rotation:

$$\mathbf{F}_{\mathbf{Total}}(t) = M\frac{d\mathbf{U}_{\mathbf{p}}(t)}{dt} \qquad (2.48)$$

$$\mathbf{T}_{\mathbf{Total}}(t) = I\frac{d\mathbf{\Omega}_{\mathbf{p}}(t)}{dt} \qquad (2.49)$$

The values needed in this calculation are the mass of the particle M, the moment of inertia of the particle I, the velocity of the particle $\mathbf{U_p}$, and the angular velocity of the particle $\mathbf{\Omega_p}$. These differential equations can be solved using a variety of numerical methods such as the Euler method or higher-order Runge-Kutta methods. With these equations, the motion of a particle can be simulated in the lattice Boltzmann method.

When considering a moving particle, the treatment of those nodes considered inside the particle must be considered. In addition, a situation where a particle moves over a node that was once outside the particle and now is inside the particle must also be addressed. Similarly, one must consider what happens when a node that was previously inside a particle is now inside the flow regime. Ladd & Verberg (2001) treat the particle as a solid shell of given mass and inertia filled with fluid that has the same density as the bulk fluid. The mass of the shell is used to calculate changes in the particle velocity due to hydrodynamic forces (Equation 2.48), while the effects of the interior fluid on the inertia of the particle must be considered. Ladd and Verberg report that the dynamic behavior of these particles does not differ significantly from the dynamics of solid particles. However, the main drawback to this method is that the instantaneous momentum fluctuations of these fluid-filled particles are larger than with solid particles. The advantage with this treatment is that the interior and exterior nodes are both treated the same since they both represent fluid regimes.

In contrast, Aidun et al. (1995, 1998) do not treat the interior and exterior nodes in the same manner and account differently for how nodes are treated when covered or uncovered by a moving object. Nodes inside a solid particle are not treated as fluid nodes as in Ladd's method. Mass and momentum are conserved through a redistribution of additional momentum about the entire surface of the solid particle to bounce-back PDF values or rest PDF values. Since the interior nodes are no longer considered part of a fluid regime, the force calculation at the surface is adjusted in the following manner:

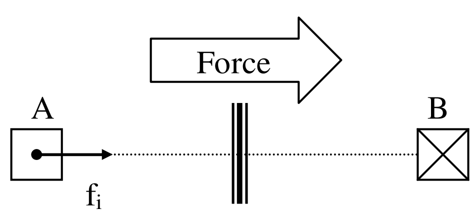


Figure 2.17 – Illustration of the Momentum-Exchange Method for Force Evaluation (fluid exists only outside of wall)

$$\mathbf{F}(\mathbf{x_W}, t+\frac{1}{2}\Delta t) = \frac{\Delta x^3}{\Delta t}\left[2 f_i^{\,c}(\mathbf{x_A}, t) - \frac{2 w_i \rho(\mathbf{u_W}\cdot\mathbf{e_i})}{c_s^2}\right]\mathbf{e_i} \qquad (2.50)$$

The wall boundary is again located halfway between the fluid node A and the solid node B and the force exerted on the wall is a function of the post-collision PDF value at node A. Note that the post-collision PDF value at node B is no longer in the calculation. The drawback with this method is that nodes must become fluid or solid in the situation of a moving particle. Aidun et al. (1995) handled the situation of solid particle covering fluid nodes by having the particle assume the momentum of the fluid node. When this occurs, a small impulse force is applied to the solid particle. In the reverse case, a fluid node that has been uncovered by a moving solid particle is given a velocity equal to the velocity at the boundary of the particle and is given a density equal to the average density of the surrounding boundary nodes. In this case as well, a small force is applied to the particle to conserve momentum. The equations for these calculations are the following:

$$\rho^{UN} = \frac{1}{N}\sum_{N}\rho_i \qquad (2.51)$$

$$\mathbf{F}^{\mathbf{UN}}(\mathbf{x_W}, t+\frac{1}{2}\Delta t) = \frac{\Delta x^3}{\Delta t}\left(-\rho^{UN}\mathbf{u_W}\right) \qquad (2.52)$$

$$\mathbf{T}^{\mathbf{UN}}(\mathbf{x_W}, t+\frac{1}{2}\Delta t) = \left(\mathbf{x_W} - \mathbf{x_{CM}}\right)\times\mathbf{F}^{\mathbf{UN}}(\mathbf{x_W}, t+\frac{1}{2}\Delta t) \qquad (2.53)$$

The density of an uncovered fluid node ρ^{UN} is the average of the densities of the surrounding N nodes. The force on a particle due to uncovering a fluid node $\mathbf{F}^{\mathbf{UN}}$ is a function of the momentum of the new uncovered fluid node, which has been given the velocity at a close boundary point. The torque on the particle $\mathbf{T}^{\mathbf{UN}}$ is simply a function of the force $\mathbf{F}^{\mathbf{UN}}$. The force and torque resulting from the covering of a fluid node also conserves momentum and is similarly found, except the density and velocity of the fluid node is already known and not calculated. It should be noted that these forces and torques are added to the other forces and torques summed around the boundary in order to obtain a $\mathbf{F}_{\mathbf{TOTAL}}$ and $\mathbf{T}_{\mathbf{TOTAL}}$ in Equations 2.48 and 2.49.

In other published studies, researchers have done lattice Boltzmann simulations using the momentum-exchange force method on particles in suspension with good results. Qi (1999, 2000) has studied 2D and 3D suspensions with spherical and non-spherical (rectangular) particles. Qi does allow for fluid inside the node as Ladd (1994a) did and noted the disadvantage of not being able to handle a solid density less than the fluid density. Qi conserves momentum in a similar way to Aidun et al. (1995) by adding a force and torque when uncovering and covering nodes. The results of these studies by Qi were consistent with independent finite-element results and other experimental results. A study done by ten Cate et al. (2002) used a similar technique to Qi, but with a slightly different force correction. The lattice Boltzmann simulations of a settling sphere in silicone oil agreed well with experimental results. Specifically, the transient behavior of the settling sphere and the motion of the fluid were effectively captured by the lattice Boltzmann simulations. Lastly, Feng & Michaelides (2002) used the momentum exchange force method to simulate the shear flow around a particle or particles near the bottom of a channel. The simulation calculated the drag and lift forces felt by the particle and also investigated the forces due to the interaction of suspended and stationary particles. These studies and others have validated the momentum-exchange method in practice.

An alternative way to calculate force on a boundary in the lattice Boltzmann method was presented by He & Doolen (1997). This method involves the integration of the stress tensor on the surface of the boundary. In this particular study, the stress on the surface of a 2D cylinder was integrated in order to get the drag force and drag coefficient. The components of the stress tensor were calculated using finite-difference equations and velocities. The stress-integration method was convenient for their study as they adapted the lattice Boltzmann method for curvilinear coordinates, which allowed the grid to be fitted to the surface. This allowed easy computations of the stress tensor and convenient surface normals. However, for situations where boundaries are not aligned with the grid, the computations can be more complicated. A study by Mei et al. (2002) evaluated the stress-integration method and compared it to the momentum-exchange method for force evaluation in the lattice Boltzmann. Several test cases were evaluated including 2D flow past a cylinder and 3D flow past a sphere. Mei et al. concluded that while both the momentum-exchange method and stress-integration method give similar results, the momentum-exchange method is easily implemented while the stress-integration method requires more effort to implement. The errors in the momentum-exchange method are inversely proportional to the resolution, while there are large fluctuations seen in the stress-integration method. Stress-integration also must process PDF values through extrapolation or interpolation while momentum-exchange requires little processing. Mei et al. recommended the use of the momentum-exchange method for its simplicity, accuracy, and robustness.

The last concern when dealing with particle dynamics in the lattice Boltzmann occurs when particles approach each other or come in contact. In general, when two spheres approach each other in a fluid regime a repulsive force is generated due to the large pressures that develop in the small gap between them. This force can be calculated by lubrication theory and has to be taken into account in particle simulations. Several methods to account for this have been presented in the literature. Nguyen & Ladd (2002), ten Cate et al. (2002), and Ding & Aidun (2003) have used lubrication theory to calculate forces when the resolution of the lattice Boltzmann grid can no longer resolve the distance between objects. In the study by ten Cate et al., the lubrication force is calculated explicitly and applied to a sphere in close proximity to a wall. Equation 2.54 is used for this calculation.

$$\mathbf{F_W} = -6\pi\mu r_p u_\perp \left(\frac{r_p}{h} - \frac{r_p}{\Delta_0} \right) \qquad (2.54)$$

The lubrication force $\mathbf{F_W}$ acting on the sphere along the direction normal to the wall is in part a function of the dynamic viscosity μ, the radius of the particle r_p, the component of the velocity of the sphere perpendicular to the wall $u_\perp$, and the gap between the sphere and the wall h. In addition, Δ_0 is a distance chosen to be equal to the lattice spacing Δx. A similar formulation appeared in the review by Ladd & Verberg (2000), but more recently Nguyen & Ladd (2002) have published a lubrication correction method for lattice Boltzmann simulations with multiple particles. The lubrication force between two particles is the calculated using Equations 2.55 and 2.56.

$$\mathbf{F_l} = -6\pi\mu \frac{a_1^2 a_2^2}{(a_1 + a_2)^2} \left(\frac{1}{h} - \frac{1}{h_N} \right) \mathbf{U_{12}} \cdot \hat{\mathbf{R}}_{\mathbf{12}}, \quad h < h_N \qquad (2.55)$$

$$\mathbf{F_l} = 0, \quad h > h_N \qquad (2.56)$$

The lubrication force $\mathbf{F_l}$ is in part a function of the particle radii a_1 and a_2, the distance between the particles h, the relative velocity between particles $\mathbf{U_{12}} = \mathbf{U_1} - \mathbf{U_2}$, and the unit vector for $\mathbf{R_{12}}$ equals $\mathbf{R_{12}} / |\mathbf{R_{12}}|$. The parameter h_N is a cutoff distance determined by Nguyen & Ladd (2002) for a particle size and fluid viscosity and serves as a correction factor. When h is less than h_N, then a lubrication force is valid and calculated with Equation 2.55. However when h is greater than h_N, the lubrication force is zero. Nguyen & Ladd used this correction factor to bring the simulation data for lubrication force in line with theoretical predictions.

Ding & Aidun (2003) have recently published a study on simulation of suspended particles in near contact. This extension of the previous work by Aidun et al. details the treatment of nodes within solid particles, termed "virtual" nodes because they contain "virtual" fluid, when two particles come within less than one lattice distance with no fluid node between them. This treatment adds rules to adding momentum to particles based on virtual node PDF values at the boundary of the particles. A second component to this extension documents the lubrication forces in these situations. Total lubrication force is calculated as a summation of elemental forces along links that span the gap between particles termed as "bridge" links. The results presented include two 3D spheres coming in close contact in a channel as well as a 2D cylinder approaching a flat wall. The lubrication forces calculated in the study compare favorably with theoretical values.

The present study will take these approaches to particle dynamics and extend them based on the boundary treatment chosen. The most natural fit for boundary treatments built on bounce-back and interpolation is the momentum-exchange method. For a moving construct in the bioreactor environment, nodes that change from solid to fluid will need special attention. The lubrication forces will also be carefully considered so to accurately simulate the bioreactor fluid dynamics.

2.2.7. Grid Refinement

There are a number of situations where a non-uniform grid is desirable when performing computations. A non-uniform grid is useful when there are areas in a grid where the details of the flow are of primary concern versus other non-vital areas. In addition, boundary fitted grids facilitate the calculation of stresses and pressure distributions on a surface. A drawback of the lattice Boltzmann method is that the original formulation of the method calls for a regular or evenly spaced grid. Several researchers have explored the possibility of implementing the lattice Boltzmann method on different types of non-uniform grids. Researchers must consider how the lattice Boltzmann is modified in order to accommodate a coarsened or refined grid.

A study published by He et al. (1996) used non-uniform grids in the lattice Boltzmann method by adding an additional interpolation step. Collision and streaming occur at nodes as normal, but the distances between nodes are no longer uniformly the lattice spacing Δx. The interpolation step occurs after collision and streaming in order to find the PDF values at nodes. This method is based on the idea that the PDF values at a location is space can always be interpolated from PDF values at neighboring locations because a PDF in space is a sufficiently smooth function. A sufficiently smooth function has continuous derivatives up to a desired order, in this case the order of interpolation. This implies that for every set of particles that travel with a velocity as defined by the lattice Boltzmann model, a smooth surface exists that fits those PDF values. In other words all the PDF values corresponding to the $\mathbf{e_1}$ direction in the D2Q9 model are part of a smooth function representing f_1 values over the entire 2D space. In the study, a non-uniform 2D rectangular grid was used to illustrate the method. Consider Figure 2.18, an illustration of a non-uniform grid.

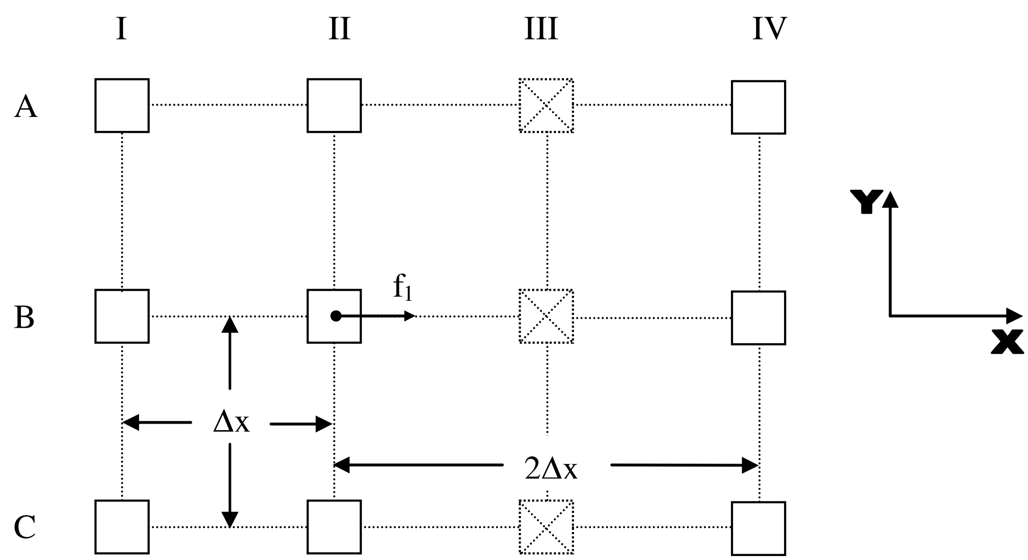


Figure 2.18 – Illustration of One Type of Non-Uniform Grid

The figure depicts a grid that is square between columns I and II, but then becomes rectangular between columns II and IV. There are no nodes in column III and the effective distance between the nodes in columns II and IV is $2\Delta x$. Now consider the PDF values (f_1 values) corresponding to the $\mathbf{e_1}$ direction. Collision takes place at each node, but after the streaming the particles traveling in the $\mathbf{e_1}$ direction from column II will end up at a location corresponding to column III. This leaves unknown PDF values at column IV. He et al. proposes the interpolation or reconstruction of values at column IV using linear or quadratic interpolation along the links connecting the nodes. In this case, a PDF value (f_1) at node B-IV will be interpolated using f_1 values along row B. The case of a 2D channel with a sudden expansion is analyzed with results that compare favorably with experimental results. In general, this method allowed for the coarsening of a grid given a lattice spacing of Δx. In contrast, most researchers have explored the refinement of a grid by placing a nested grid within the coarser main grid.

In refining lattice Boltzmann grids, researchers have proposed placing a grid refined by some factor N within the main grid. So for example, a refinement grid where N = 2 has a lattice spacing that would be 1/2 of the main grid. The figure below is an illustration of a refinement grid, or also termed sub-grid, placed within a larger grid. In reference to the figure, nodes B, D, and F are along the boundary of the sub-grid and nodes A, B, E, and F are nodes that exist in both the larger main grid and smaller sub-grid.

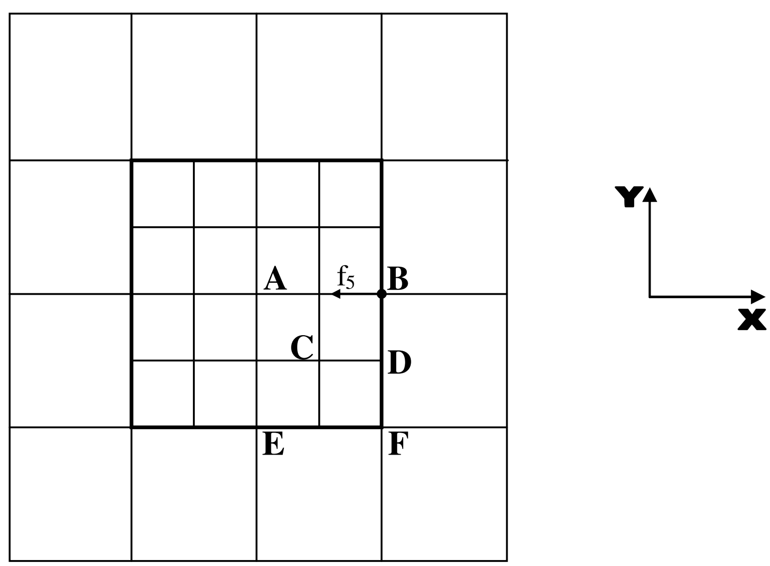


Figure 2.19 – Illustration of an Embedded Sub-Grid

When placing a refined grid within a coarse grid, there are several issues that need to be addressed. The first issue is the function of the sub-grid with respect to the lattice Boltzmann method. The second issue is the calculation of the parameters of the sub-grid relative to the larger grid. The last issue is the interface between the sub-grid and the main grid at the boundaries. In general, a sub-grid will exist over the coarse grid and the interaction between the two must be defined. An initialization of the sub-grid involves interpolating PDF values at each node based on PDF values in the main grid. For example, in the figure PDF values on the main grid at A, B, E, and F would be used to interpolate PDF values at C and D. The computation proceeds at large time steps Δt^{MG} for the main grid. The sub-grid flow is calculated at smaller time steps $\Delta t^{SG} = \Delta t^{MG}/N$ in order to advance to the same time level. An issue arises when nodes at the boundary such as node D require PDF values that stream into the sub-grid. These values are typically interpolated in space and time from PDF values at surrounding nodes. While this is a general procedure in how sub-grids are implemented, researchers have presented slight variations.

Filippova & Hänel (1998a, 2000) proposed a method that coupled the solutions found on the sub-grid and main grid. The sub-grid utilizes a different relaxation parameter with different lattice spacing to maintain similarity in viscosity. The sub-grid must also scale the non-equilibrium portion of PDF values when crossing into and out of the sub-grid. This is one of the distinctive features of the method in that information from the sub-grid will flow out to the coarse grid by scaling the PDF values. The equations to do this are the following:

$$\tau_{SG} = \frac{1}{2} + N\left(\tau_{MG} - \frac{1}{2}\right) \tag{2.57}$$

$$f_i^{SG} = f_i^{eq,MG} + \frac{1}{N}\frac{(\tau_{SG}-1)}{(\tau_{MG}-1)}\left(f_i^{MG} - f_i^{eq,MG}\right) \tag{2.58}$$

$$f_i^{MG} = f_i^{eq,SG} + N\frac{(\tau_{MG}-1)}{(\tau_{SG}-1)}\left(f_i^{SG} - f_i^{eq,SG}\right) \tag{2.59}$$

The relaxation parameter for the sub-grid τ_{SG}, is calculated from the refinement factor N, and the relaxation parameter for the main grid τ_{MG} in Equation 2.57. The PDF values that stream into the sub-grid f^{SG} are scaled in Equation 2.58 from the PDF values on the main grid f^{MG} and the relaxation parameters. In Figure 2.19, for a D2Q9 model this equation would scale the sub-grid f_5 value at node B from the f_5 values at node B on the main grid. The PDF values that stream into the main grid are scaled in Equation 2.59 from the PDF values on the sub-grid. The accuracy of this method was demonstrated by test cases involving flow over a circular cylinder at various Reynolds numbers.

Lin & Lai (2000) reported on a composite grid method for the lattice Boltzmann that unlike the Filippova & Hänel method only passes information into the sub-grid domain. Lin & Lai emphasized more CPU time saved when implementing the method and explained the interpolation done on the boundary in intermediate time steps in more detail. They noted that by applying finer grids in areas of interest allows the less computed time steps on the main grid to reach a certain time level because Δt is larger. In addition, the number of nodes is reduced by a factor of N^2 in 2D and N^3 in three dimensions. The combined savings in computational time from less time steps and less nodes are significant. Lin & Lai also detailed the method by which they interpolate at the sub-grid boundary. Consider Figure 2.19 and the following illustration of the time steps when N = 2.

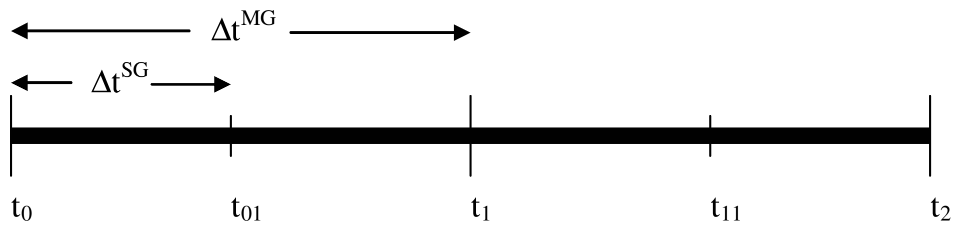


Figure 2.20 – Illustration of Time Steps for a Sub-Grid Implementation

At the start of the sub-grid computation, time t_0, the sub-grid is initialized with PDF values at every node. This is done using interpolation and PDF values on the main grid. The computation proceeds on the main grid to time t_1. Now that values on the main grid are known for both t_0 and t_1, the sub-grid computation can commence to time t_{01}. On the sub-grid, collision and streaming take place. However, at the sub-grid boundary there is no new information for certain PDF values because there are no nodes to stream these PDF values to boundary nodes. Again consider the PDF value f_5 for boundary node B on the sub-grid. In this case since $f_5(\mathbf{x_B},t_0)$ and $f_5(\mathbf{x_B},t_1)$ are known from the main grid, the value $f_5(\mathbf{x_B},t_{01})$ can be interpolated in time linearly. For the PDF value f_5 at boundary node D, more information is needed. A linear interpolation in space and time needs the values $f_5(\mathbf{x_B},t_0)$, $f_5(\mathbf{x_B},t_1)$, $f_5(\mathbf{x_F},t_0)$, and $f_5(\mathbf{x_F},t_1)$ in order to calculate $f_5(\mathbf{x_D},t_{01})$ on the sub-grid. This process is done after the collision and streaming on the sub-grid and allows for a new velocity and pressure to be calculated at the boundary nodes. The sub-grid then advances to t_1 where interpolation is again needed at boundary nodes. Using this method, Lin & Lai simulated lid-driven cavity flow at a Reynolds number equal to 1000. Their results compared favorably with previous independent studies.

There have been other methods proposed by Kandhai et al. (2000) and by Yu et al. (2002). The method by Kandhai et al. extends the method by further nesting grids within sub-grids. Instead of scaling the PDF values like Filippova & Hänel, Kandhai et al. proposed making the velocities in the model $\mathbf{e_i}$ grid dependent. This was done by introduction a parameter that also allows for the speed of sound and kinematic viscosity to be equal on all grids. Results for a 2D Taylor vortex problem were presented with good agreement with analytical values. Yu et al. propose a multi-block method in which the sub-grid does not just overlap the main grid, but replaces it. The boundary of the sub-grid and main grid overlap for one lattice spacing on the main grid, but there are no coexisting nodes beyond the boundary. Results for lid-driven cavity flow showed good velocity agreement with independent results. In addition, smooth pressure and shear stress contours are noted at the interface between grids. Simulation of flow over an airfoil also generated comparable results with data reported by other researchers. It is stressed that the method maintains mass conservation and stress continuity through cubic spline interpolation at the boundary and proper scaling of PDF values using the equations of Filippova & Hänel.

Methods for the refinement of the grid in the lattice Boltzmann have been presented in the literature. Grid refinement will directly contribute to saving CPU time on non-vital parts of the flow, thus making the method more efficient. Correctly implementing a flexible grid refinement method for the bioreactor problem will require considering the approaches previously presented in literature. However, the most important features of the method including consistency of parameters across the grids and placing the sub-grid will be of primary concern in this study.

2.2.8. Accuracy

The accuracy of the lattice Boltzmann method has been reported in the literature over the past several years. Some of the key issues that researchers address include the overall accuracy of the method, how boundary treatments affect the accuracy of the method, and the criteria at which the method converges. In order to verify the accuracy of their particular implementation of the lattice Boltzmann method, researchers have compared their results with various other methods. An overview of some of these issues is presented in this portion of the literature review.

Accuracy of the lattice Boltzmann method was briefly explored by Maier & Bernard (1997) who noted that in general, the accuracy of the method is moderated by several factors including spatial resolution, the Mach number, and the lattice mean free path. The accuracy is also practically affected by the particular implementation of the method and the boundary treatments used. Specifically, the lattice Boltzmann is known to have a second-order rate of spatial convergence away from the boundaries, but accuracy at the boundaries depends on the treatment used. Maier & Bernard also found that the maximum relative error can be directly related to the product of the Mach number and Knudsen number for flows where second-order boundary treatments are used. The Mach number is the ratio of velocity magnitude to the speed of sound while the Knudsen number is the ratio of mean free path to the macroscopic length scale. These results gave an overall view of accuracy in the lattice Boltzmann leaving more research to be done on more specific implementations. In a more comprehensive study, Lai et al. (2001) examined the accuracy and efficiency of the lattice Boltzmann method for steady state flows. Lai et al. report that the overall accuracy of the method is between first-order and second-order depending on the boundary treatment used. The method by which overall accuracy was determined was based on comparing solutions on refined grids. Equation 2.60 can be used to estimate the order of the overall accuracy *n*.

$$n \approx \frac{\log\left[\left(\frac{\sum\left|\phi_{2h}-\phi_{4h}\right|/N}{\sum\left|\phi_{h}-\phi_{2h}\right|/N}\right)\right]}{\log(2)} \qquad (2.60)$$

This formula is derived from simple error analysis of solutions when using grid spacings h, 2h, and 4h which represent fine, medium and coarse grids. The symbol ϕ represents a dependent variable which was chosen as the streamwise component of velocity in this case, and N represents the total number of grid points compared. In this study, the order of overall accuracy for lid-driven cavity flow was found to be 1.41 for the bounce-back boundary treatment and 2.08 for the MLS boundary treatment. Similar results were obtained for 2D and 3D flow over a backward facing step. Lai et al. noted that these results support the fact that the lattice Boltzmann method is an accurate alternative method for solving incompressible fluid flows, but that is still more computationally expensive then more advanced conventional Navier-Stokes solvers.

More commonly, researchers have reported on accuracy results for the implementation of a specific boundary treatment. There are various ways to evaluate the error and illustrate how the error decays with an increase in grid refinement. Gallivan et al. (1997) evaluated the accuracy of the bounce-back boundary treatment in relation to the hydrodynamic boundary treatment of Noble and a finite-difference calculation. The root mean square norm of the error and maximum error are defined by Gallivan et al. as the following:

$$\text{error norm} = \left[\frac{\sum \left[\left(u_{LB} - u^* \right)^2 + \left(v_{LB} - v^* \right)^2 \right]}{\sum \left[\left(u^* \right)^2 + \left(v^* \right)^2 \right]} \right]^{1/2} \tag{2.61}$$

$$\text{error max} = \max \left[\left(u_{LB} - u^* \right)^2 + \left(v_{LB} - v^* \right)^2 \right]^{1/2} \tag{2.62}$$

The error norm and max error are functions of the components of velocity of the current solution using the lattice Boltzmann u_{LB} and v_{LB} and the solution from calculations taken as accurate u^* and v^*. These values are taken at all grid points in the flow domain for flow around an octagonal and circular cylinder. The errors are plotted against the lattice spacing Δx and it was shown that the slope for the error norm of the bounce-back treatment was about 1 while the hydrodynamic boundary treatment slope was about 2. This illustrated the first-order and second-order convergence of the two treatments. Verberg and Ladd (2002) evaluate the continuous bounce-back boundary treatment by evaluating the error in simulating 2D channel Poiseuille flow. The RMS error across the channel was calculated as a function of the number of grid points across the channel L, the current solution using the lattice Boltzmann v_{LB}, the exact solution v_{EXACT}, and the exact solution at the centerline v_c.

$$\varepsilon(L) = \frac{1}{v_c} \left[\frac{1}{L} \sum_{\mathbf{r} \in L} \alpha \left(v_{LBE}(\mathbf{r}) - v_{EXACT}(\mathbf{r}) \right)^2 \right]^{1/2} \tag{2.63}$$

Verberg & Ladd plot this error against L and note that there is second-order convergence as L increases. In contrast, the L_2 error norm was used by Mei et al. (1999) to evaluate the solution in a pressure driven channel flow. The formula for the relative L_2

norm of the error is a function of the height of the channel H, and the velocity solutions across the channel u_{LB} and u_{EXACT}.

$$E_2 = \frac{\left[\int_0^H \left(u_{LB}(y) - u_{EXACT}(y)\right)^2 dy\right]^{1/2}}{\left[\int_0^H \left(u_{EXACT}(y)\right)^2 dy\right]^{1/2}} \qquad (2.64)$$

This error was plotted against the channel height H and second-order convergence was observed. While the error analysis may look slightly different, the concept in demonstrating second-order convergence is basically the same. However, the convergence of the lattice Boltzmann solution must still be addressed.

The convergence criteria for a lattice Boltzmann simulation have not been mentioned in the literature often. Two similar criteria have been presented by Filippova & Hänel (2000) and He et al. (1996). These criteria are presented in the following equations:

$$\sum_i \frac{\left\|\mathbf{u}(\mathbf{x_i}, t+\Delta t) - \mathbf{u}(\mathbf{x_i}, t)\right\|^2}{\left\|\mathbf{u}(\mathbf{x_i}, t+\Delta t)\right\|^2} \leq 1.0x10^{-7} \qquad (2.65)$$

$$\frac{\sum_{i,j}\left\|\mathbf{u}(\mathbf{x_{i,j}}, t+\Delta t) - \mathbf{u}(\mathbf{x_{i,j}}, t)\right\|}{\sum_{i,j}\left\|\mathbf{u}(\mathbf{x_{i,j}}, t)\right\|} \leq 1.0x10^{-6} \qquad (2.66)$$

The convergence criteria above are based on the L_2 norms of the relative error associated with the velocity vectors. The L_2 norm is denoted by $\|\cdot\|$. The differences are subtle between the two criteria, but one uses the solution at the current time step as reference rather than using the solution at the last time step. He et al. (1996) noted at that time the convergence in their simulations took about 50,000 time steps. In general, the number of time steps will depend on the implementation of the lattice Boltzmann and the problem being simulated. Another convergence criterion is presented by Zou et al. (1995). This criterion is stated in Equation 2.67.

$$\frac{\sum_{i,j}\left[\ \left|u(\mathbf{x_{i,j}},t+\Delta t)-u(\mathbf{x_{i,j}},t)\right|+\left|v(\mathbf{x_{i,j}},t+\Delta t)-v(\mathbf{x_{i,j}},t)\right|\ \right]}{\sum_{i,j}\left[\ \left|u(\mathbf{x_{i,j}},t)\right|+\left|v(\mathbf{x_{i,j}},t)\right|\ \right]}\leq \Delta x\cdot \text{tolerance} \qquad (2.67)$$

In this study, the tolerance is chosen as 10^{-12} for this criterion. Zou et al. (1995) also mentioned that 10^{-8} could be chosen as the tolerance as there was no observed difference in the error in the velocity, comparatively. This criterion is different from the first two presented in that it does not compute an L_2 norm. Therefore, researchers are prone to choose different accuracy measures as well as convergence criterion. An issue that must be monitored is the fluctuation of the value of the convergence criterion. Typically, the value of the convergence criterion will not monotonically decrease so careful consideration must be given to when the simulation has converged. The issue of convergence is not widely reported on in the literature, but must be carefully considered.

Lastly, in reviewing the literature on how the accuracy of the lattice Boltzmann method has been determined, the solutions that the calculations are compared against must be assessed. In most cases where boundary treatment accuracy is being examined, a simple case such as Couette or Poiseuille channel flow is used as a benchmark. This is convenient because an analytical solution exists and the error analysis is made simple. However, the target problem being simulated is always more complex and comparative solutions must reflect that complex nature. In the literature, some comparisons are made to finite difference solutions computed by the researchers themselves or to previous data. Comparing simulations with experimental data such as the PIV results of a settling sphere by ten Cate et al. (2002) has also been seen in the literature. The key issue is how researchers qualify the agreement of their simulation data with the reference data which is viewed as an accurate solution. In most cases the methodology from researcher to researcher is different without a standard to compare. One of the objectives of the current study is to make that methodology transparent, easily understood, and easily verified. The accuracy of the lattice Boltzmann method should be apparent from the results if presented in a compelling way.

2.3. Summary

This chapter offers a detailed review of research involving both the RWV bioreactor and the lattice Boltzmann method. An understanding of the research involving the RWV bioreactor is necessary in appreciating the critical issues for this particular problem. These issues include the motion of the tissue construct and shear stresses that are felt by the construct and consequently how these affect the growth of the construct. The research published involving the lattice Boltzmann method is extensive and includes examination of boundary treatments, grid refinement, and moving boundaries. However, there is no study that examines the accuracy of these components of the lattice Boltzmann as a whole. This research attempts to tie these lattice Boltzmann components into one model that can accurately describe the RWV bioreactor environment.

Chapter 3

Methodology

The accuracy of the implementation of the lattice Boltzmann method for the RWV bioreactor problem will determine the usefulness of the model. If the implementation of the method is both accurate and innovative, the methodology used in this study will aid both the study of the RWV bioreactor and the further development of the lattice Boltzmann method. This chapter will address the technical implementation aspects of the method with an emphasis on the models used, a new boundary treatment formulation, the modeling of particle dynamics, and the use of grid refinement. The chapter will also address the practical implementation aspects of the lattice Boltzmann method by examining software and hardware with respect to serial and parallel implementation. The methodology presented here can provide a starting point for more advanced RWV simulations that can further address details such as mass transfer and porosity of constructs. It also provides a framework for further advancement of accurate lattice Boltzmann models.

3.1. Implementation of the Lattice Boltzmann Method

Many aspects of the lattice Boltzmann method and its implementation have been explored in the research literature. The implementation presented is based on the most widely used simplification of the lattice Boltzmann equation, the LBGK equation, and a common choice for the equilibrium particle distribution function. While there have been numerous improvements to these basic equations such as multi-relaxation time models

(Lallemand & Luo 2000) or accelerated implementations (Filippova & Hänel 2000), this study will aim to verify the accuracy of the basic equations only. This allows flexibility in applying current and future improvements proposed in the literature that may speed up computation time or increase stability. The specific aspects of the implementation discussed are the boundary treatment, particle dynamics, and the grid refinement. The formulation of these aspects is a product of methods discussed in previous literature and the innovation of these methods. The accuracy of the model will depend on the implementation of these aspects.

3.1.1. 2D and 3D Models

The models used in this study will be based on the LBGK equation (Equation 2.2). As noted, the most common 2D model presented in the literature is the D2Q9 model. This will be the model used for all 2D simulations in this study. This model assigns nine velocities to the particles which move on a square lattice. For these nine velocities there are nine velocity directions assigned, but only three velocity magnitudes (velocity magnitude 0 = 0, velocity magnitude I = 1, and velocity magnitude II = √2). While there are variations to velocity designations for the D2Q9 model in literature, the velocities in this study are designated in the following way:

$$\begin{aligned}
&\text{Velocity Magnitude I}\\
&\mathbf{e}_\mathrm{i} = \left(\cos\left(\frac{\pi(i-1)}{4}\right), \sin\left(\frac{\pi(i-1)}{4}\right)\right)\frac{\Delta x}{\Delta t} \qquad \mathrm{i} = 1, 3, 5, 7\\
&\text{Velocity Magnitude II}\\
&\mathbf{e}_\mathrm{i} = \sqrt{2}\left(\cos\left(\frac{\pi(i-1)}{4}\right), \sin\left(\frac{\pi(i-1)}{4}\right)\right)\frac{\Delta x}{\Delta t} \qquad \mathrm{i} = 2, 4, 6, 8\\
&\text{Velocity Magnitude 0}\\
&\mathbf{e}_\mathrm{o} = 0
\end{aligned}$$

Figure 3.1 – Velocity Designations for the D2Q9 Model

For this study the equilibrium particle distribution function is chosen to follow the form of Equation 2.8. The kinematic viscosity and pressure calculations in this study follow the form of Equations 2.9 and 2.10, respectively. Again, the reasons for choosing this model were that it was the most common seen in literature and that future improvements seen in literature can be applied at a later date.

The 3D model chosen for this study is D3Q19 model. This model was chosen because of its reported performance in the literature. The D3Q19 model was found to not have the checkerboard invariants seen in the D3Q15 model. Also, the memory requirements of the D3Q19 model are less than the D3Q27 model which did not show a significant increase in performance or accuracy. For the D3Q19 model, there are nineteen velocities assigned to particles moving on a cubic lattice. Similar to the D2Q9 model, there are three velocity magnitudes (0, 1, and √2) among the nineteen velocities. For this study, the velocities are assigned the following values:

Velocity magnitude I

$$\mathbf{e_1} = (1,0,0)\frac{\Delta x}{\Delta t} \qquad \mathbf{e_2} = (0,1,0)\frac{\Delta x}{\Delta t} \qquad \mathbf{e_3} = (0,0,1)\frac{\Delta x}{\Delta t}$$

$$\mathbf{e_4} = (-1,0,0)\frac{\Delta x}{\Delta t} \qquad \mathbf{e_5} = (0,-1,0)\frac{\Delta x}{\Delta t} \qquad \mathbf{e_6} = (0,0,-1)\frac{\Delta x}{\Delta t}$$

Velocity magnitude II

$$\mathbf{e_7} = (1,1,0)\frac{\Delta x}{\Delta t} \qquad \mathbf{e_8} = (-1,1,0)\frac{\Delta x}{\Delta t} \qquad \mathbf{e_9} = (-1,-1,0)\frac{\Delta x}{\Delta t} \qquad \mathbf{e_{10}} = (1,-1,0)\frac{\Delta x}{\Delta t}$$

$$\mathbf{e_{11}} = (1,0,1)\frac{\Delta x}{\Delta t} \qquad \mathbf{e_{12}} = (-1,0,1)\frac{\Delta x}{\Delta t} \qquad \mathbf{e_{13}} = (-1,0,-1)\frac{\Delta x}{\Delta t} \qquad \mathbf{e_{14}} = (1,0,-1)\frac{\Delta x}{\Delta t}$$

$$\mathbf{e_{15}} = (0,1,1)\frac{\Delta x}{\Delta t} \qquad \mathbf{e_{16}} = (0,-1,1)\frac{\Delta x}{\Delta t} \qquad \mathbf{e_{17}} = (0,-1,-1)\frac{\Delta x}{\Delta t} \qquad \mathbf{e_{18}} = (0,1,-1)\frac{\Delta x}{\Delta t}$$

Velocity magnitude 0

$$\mathbf{e_0} = 0$$

Figure 3.2 – Velocity Designations for the D3Q19 Model

The equilibrium particle distribution function chosen for the D3Q19 model is the same as the D2Q9 model (Equation 2.8) except with different w_i weighting factors. The w_i factors used for the D3Q19 model corresponding to the velocity magnitudes are $w_0 = 1/3$, $w_I = 1/18$, and $w_{II} = 1/36$. The kinematic viscosity and pressure are calculated in the same way as the D2Q9 model (Equations 2.9 and 2.10). The similarities between the D2Q9 model and the D3Q19 model facilitate implementation.

The difference in the practical implementation of these models is small. Once a lattice Boltzmann method has been programmed, the implementation of different types of models is a matter of addressing some minor details. The general characteristics and steps of the lattice Boltzmann method that must be considered in programming include grid generation, collision and streaming, boundary treatments, etc. Out of these, the grid generation will be affected the most because a 3D grid will differ from a 2D grid, while the collision and streaming processes are affected in a smaller way through minor differences in calculations. Ideally, any processes that involve boundary treatments, particle dynamics, and grid refinement will not be affected by the model chosen. This is accomplished through the flexibility of the implementation of these aspects of the lattice Boltzmann method. As an example, in the literature there are certain boundary treatments that have difficulty being extended to 3D because of the specific equations that must be solved. This inflexibility between models should be avoided. This study has attempted to develop these common aspects of the lattice Boltzmann method so that the type of model chosen will not have an adverse effect on the implementation.

3.1.2. Boundary Treatment

The boundary treatment proposed in this study results from considering several issues. The primary consideration is the development of a boundary treatment that is easily implemented in both 2D and 3D. Previous boundary treatments like the hydrodynamic, the non-equilibrium bounce-back, and the non-slip boundary treatments require the solution of a set of equations specific to the geometry of the boundary. This is an inflexible approach in that extension to 3D is difficult. A second consideration in developing this boundary treatment is the consistency of the treatment with the characteristics of the model. In the study published by He et al. (1996), the proposed method for non-uniform grids was based on a simple concept concerning the particle distribution function values. He et al. proposed that the PDF values in the model made up sufficiently smooth surfaces in space. This concept will be central in the implementation of the proposed boundary treatment. Another consideration is accuracy. The boundary treatment must not only have second-order error convergence which is seen in most treatments presented in literature, but must also produce accurate results at the boundary. Both issues must be satisfied as a requirement for a new boundary treatment. Lastly, the simplicity of the boundary treatment is considered. The calculations required for such treatments like the MLS and CBB boundary treatments are at times overly complex because they change for different boundary positions. The new boundary treatment should be simple and straight forward to implement and understand. The boundary treatment presented will address these issues in order to facilitate the simulation of the curved wall geometry of the RWV bioreactor and the moving construct inside the bioreactor.

The boundary treatment developed in the course of this research study is based on the sufficiently smooth surface concept presented by He et al. (1996) and the simplicity of the Bouzidi interpolation boundary treatment (2001). The boundary treatment is termed the multi-dimensional interpolation (MDI) boundary treatment because it uses interpolation in multiple dimensions rather than interpolation only along the path of a particle. The basic idea behind the MDI treatment is that the PDF values associated with any one particular velocity will make up a sufficiently smooth surface. At a boundary, particles will bounce-back according to momentum-exchange rules and can provide information for the PDF surface close to the boundary. The boundary can be at any position between nodes which contributes to the flexibility of the MDI treatment. When a node at a boundary requires a PDF value or when a boundary PDF value is unknown, information from that node *and* surrounding nodes are used to interpolate the PDF value. Information from these surrounding nodes makes the treatment multi-dimensional and unique from previously proposed boundary treatments. Considering the information from surrounding nodes is more consistent with the elliptic nature of the incompressible momentum equations in that the effects of perturbations are omnidirectional. Specifically, the required or unknown PDF value is interpolated from information that includes PDF values found at the boundary, values that bounce-back from the boundary, and values that have streamed away from the boundary. The interpolation is done by fitting a surface, usually higher order, to these various PDF values in order to find the required or unknown PDF value. This interpolation requires that nodes have more information in the implementation of the MDI treatment.

The implementation of the MDI treatment requires the storage and calculation of several pieces of information. First, the exact boundary position relative to nodes needs to be stored. This is required in order to get the PDF values at the boundary and the position of PDF values that bounce-back from the boundary. The PDF values at the boundary at the end of a time step are calculated from known PDF values. This calculation requires a quadratic interpolation using three PDF values and is illustrated in Figure 3.3.

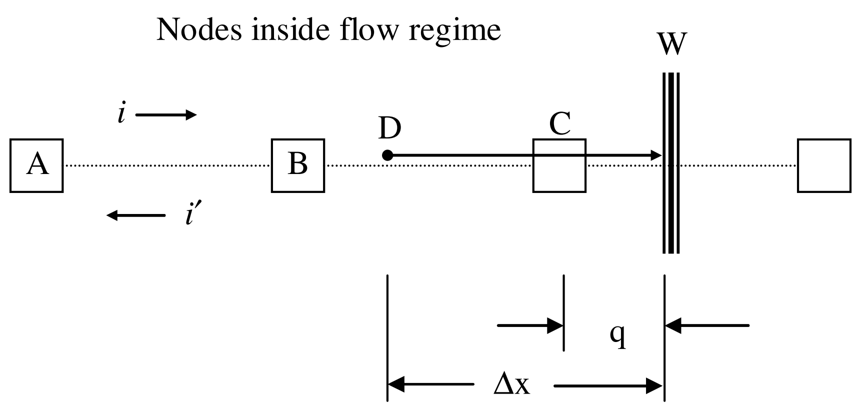


Figure 3.3 – Illustration for Calculating a PDF Value at the Wall

Given the normalized distance q of the node to the wall, the position D can be found. Note that q is normalized by the lattice spacing Δx. Position D is the position of a particle post-collision and pre-advection that will end up exactly at the wall at the end of the time step. This idea is identical to what Yu et al. (2003) proposed, but was developed independently of that research. More specifically from Figure 3.3, the post-collision PDF values $f_i(\mathbf{x_A},t)$, $f_i(\mathbf{x_B},t)$, and $f_i(\mathbf{x_C},t)$ can be used to find the post-collision PDF value $f_i(\mathbf{x_D},t)$ by applying quadratic interpolation. Then after streaming and applying momentum exchange at the wall (Equations 2.42 and 2.43) to the value $f_i(\mathbf{x_D},t)$, the wall PDF value

$f_{i'}(\mathbf{x_W},t+\Delta t)$ is found. Note that the wall PDF value is assigned the opposite direction of the original PDF values used for interpolation because it has bounced back from the wall. The position of the particles that bounce-back from the boundary is easily determined if the position of the boundary is known. Depending on the distance q from the node to the wall, the particle can end up on either "side" of the original node. Figure 3.4 illustrates these two situations.

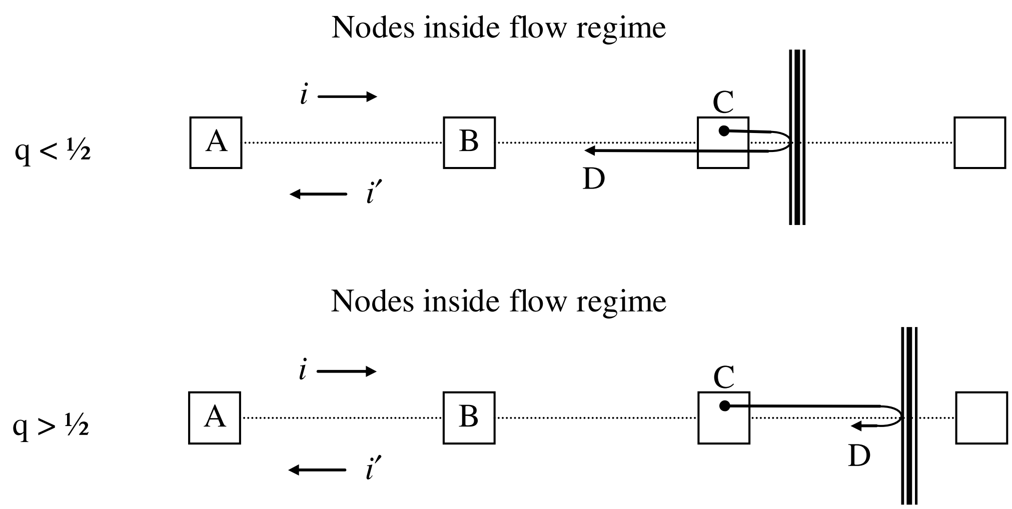


Figure 3.4 – Illustration for Calculating Position of Bounce-Back Particles

In Figure 3.4, when q < ½ the particle will end up between nodes B and C after streaming. After applying momentum exchange at the wall to the post-collision PDF value $f_i(\mathbf{x_C},t)$, the post-advection PDF value $f_{i'}(\mathbf{x_D},t+\Delta t)$ will be had at position D. Position D is easily found if q is known. In the second case when q > ½, the particle will end up between node C and the wall after streaming. Again after applying momentum exchange at the wall to $f_i(\mathbf{x_C},t)$, the post-advection PDF value $f_{i'}(\mathbf{x_D},t+\Delta t)$ at position D can be found. Once these particle positions and PDF values are known, the information can be used in the overall interpolation. The ideas presented in Figures 3.3 and 3.4 help illustrate the calculation of the PDF values needed along the boundary for the multi-dimensional interpolation.

The MDI boundary treatment tracks the exact position of the boundary at each node and interpolates an unknown PDF value using known PDF values near and at the boundary. Consider the Figure 3.5 which illustrates a situation in the D2Q9 model where a PDF value is unknown at the boundary.

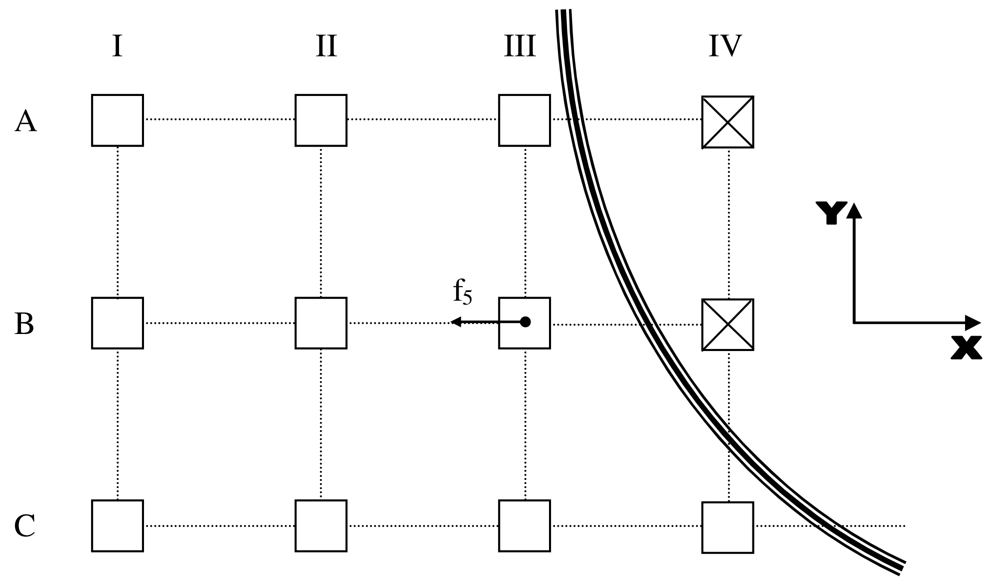


Figure 3.5 – Illustration for an MDI Boundary Treatment Example in the D2Q9 Model

The node of interest in Figure 3.5 is node BIII. Since a boundary exists between node BIII and BIV, at the end of a time step the PDF value $f_5(\mathbf{x_{BIII}},t+\Delta t)$ is unknown. For the MDI boundary treatment, information from node BIII and its nearest surrounding nodes will be used for the interpolation. In this case, nodes AII, AIII, AIV, BII, BIV, CII, CIII, and CIV are the nearest neighboring nodes and will be considered. First, nodes AIV and BIV will not be useful for the interpolation because they lie on the other side of the boundary. From the nodes that are left, they potentially can contribute streamed PDF values, bounce-back PDF values, and wall PDF values depending on their position. In this case the rest of the nodes will contribute PDF values that are associated with particles streamed with the $\mathbf{e_5}$ velocity. For example, the PDF value $f_5(\mathbf{x_{CII}},t+\Delta t)$ which streamed from node CIII (originally the post-collision value $f_5(\mathbf{x_{CIII}},t)$) will be used in the multi-dimensional interpolation. In addition to streamed PDF values, nodes AIII, BIII, and CIV will contribute bounce-back and wall PDF values because they are next to the boundary. For example at node AIII, there will be one PDF value calculated at the wall $f_5(\mathbf{x_{AIII\text{-}W}},t)$) and one PDF value that will bounce back to a position between nodes AII and AIII since q < ½ in this case. In contrast for node BIII, there will be a one PDF value calculated at the wall and one PDF value that will bounce back to a position between node BIII and the wall since q > ½. The following figure shows the post-advection position of the particles with PDF values that will be used for the interpolation.

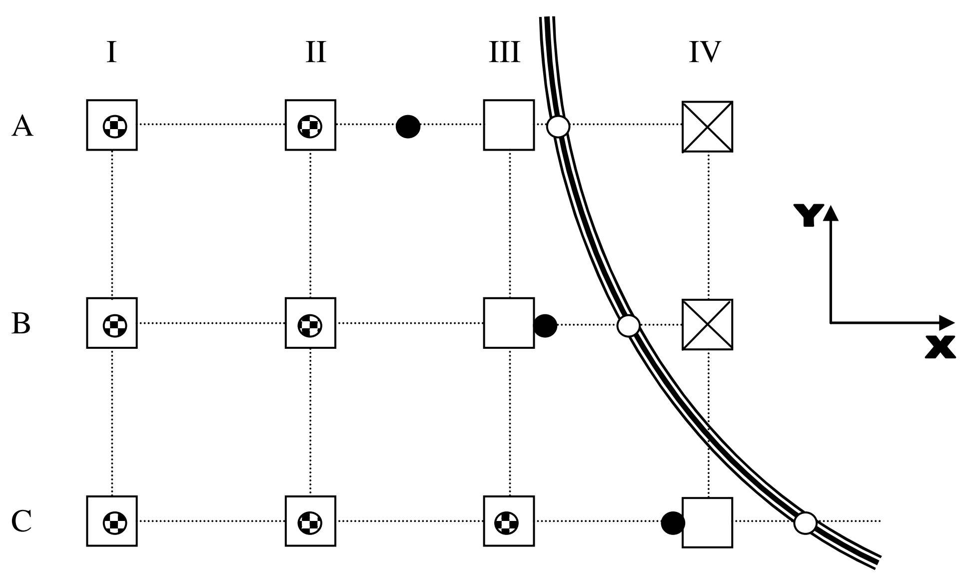


Figure 3.6 – Particle Positions for the $\mathbf{e_5}$ Velocity Case

In Figure 3.6, the hollow circles represent the position of particle PDF values calculated at the walls. Again, these values are calculated using post-collision, pre-advection PDF values (Figure 3.3). The solid circles represent the position of bounce-back PDF values from the wall (Figure 3.4). The checkered circles represent particles that streamed with the $\mathbf{e_5}$ velocity that simply end up at a neighboring node. In this case, there are 13 PDF values that are used in the multi-dimensional interpolation: 7 streamed, 3 bounce-back, and 3 wall PDF values. The position of these particles and their associated PDF values are used to fit a quadratic surface for that patch of space near the boundary. This fitted surface represents a small portion of the sufficiently smooth surface associated with all the particles streaming with the $\mathbf{e_5}$ velocity. From the fitted surface, the unknown PDF value at node BIII $f_5(\mathbf{x_{BIII}},t+\Delta t)$ can be found. Having more PDF values along the boundary allows for a better approximation of the smooth surface and thus a better approximation of the unknown PDF value. This is the primary strength of the MDI treatment. Previous boundary treatments have only considered interpolation along the link or path of the particle of interest. For example, the unified boundary treatment by Yu would only consider the 3 PDF values along the B row (2 streamed and 1 at the wall) to approximate $f_5(\mathbf{x_{BIII}},t+\Delta t)$. The extra information provided by the other 10 PDF values in the MDI treatment is critical for a faithful representation of what is happening at the boundary. A better illustration of this can be seen if you consider the same situation except for the unknown PDF value $f_6(\mathbf{x_{BIII}},t+\Delta t)$ at node BIII. Recalling that a particle with the $\mathbf{e_6}$ velocity streams down and to the left in this particular D2Q9 model (Figure 2.6), the resulting particle positions for this situation are illustrated in Figure 3.7. In this case, there are 15 PDF values used in the interpolation: 7 streamed, 4 bounce-back, and 4 wall PDF values. A surface fitted to these 15 PDF values should result in a better approximation for the unknown PDF value $f_6(\mathbf{x_{BIII}},t+\Delta t)$ because of the extra information provided by the bounce-back particles from nodes AIII, BIII, and CIII. Boundary treatments that don't consider this information or just consider information along the link or path of interest are limited in representing what is happening in the space around the node of interest. For example, the unified boundary treatment would only consider the streamed values at nodes DI and DII, and the wall PDF value to interpolate the unknown PDF value $f_6(\mathbf{x_{BIII}},t+\Delta t)$.

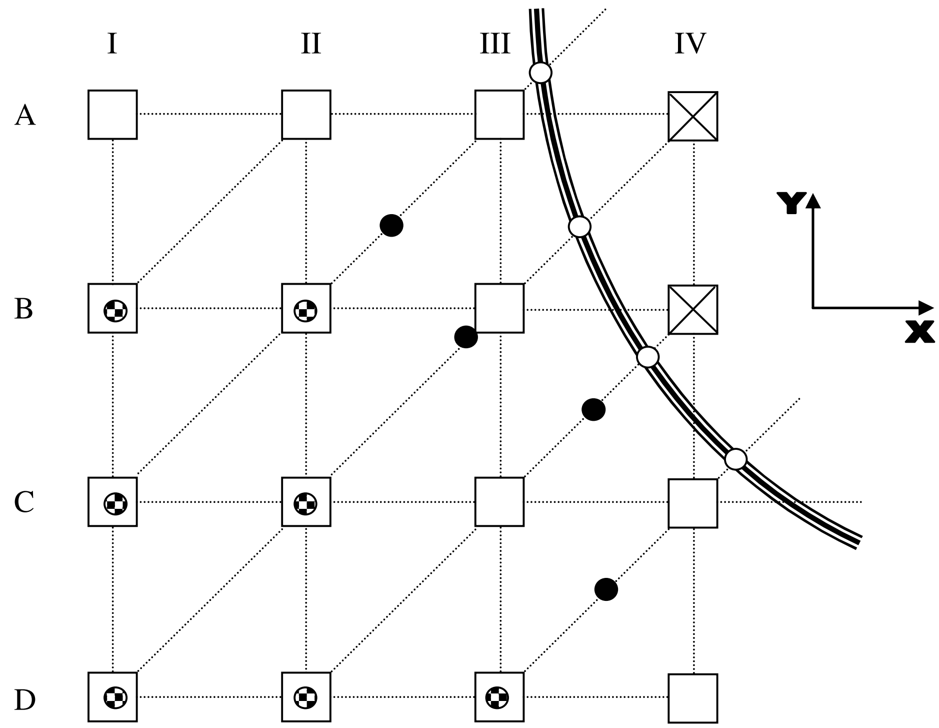


Figure 3.7 – Particle Positions for the $\mathbf{e_6}$ Velocity Case

The use of more information for interpolation is the strength of the MDI boundary treatment. The concepts behind the MDI treatment are simple and straightforward to understand. The smooth surface concept for the PDF values provides the physical reasoning and connection of the treatment with the lattice Boltzmann model. The MDI treatment is also very flexible in that the boundary can be placed at any position between nodes and that the method can be easily extended to 3D. Since the boundary can be placed at any position, a translating and moving boundary can also be handled by the MDI treatment by simply applying the momentum-exchange rules. An extension to 3D simply requires more PDF values to be calculated near a boundary and a different interpolation function for fitting iso-surfaces in space. Calculations for the MDI treatment are simple and are not dependant on rules concerning boundary position or on ad-hoc parameters. While all of these characteristics of the MDI treatment are desirable, the real test of the quality of a boundary treatment involves its accuracy. The accuracy of the MDI treatment will be addressed and tested in this study through several numerical simulations. These simulations will include simple cases and cases involving the RWV bioreactor.

The main drawbacks or disadvantages of the MDI treatment involve aspects of the implementation. One drawback is the additional computational time necessary to actually fit a surface to the given points. This computation is not trivial and is done with a routine from the software package MATLAB which is discussed later in this chapter. Since this fitting is done for every unknown PDF value at each boundary node, this can add a significant amount of overhead to the code. A second computation that adds to the overhead of the code is the quadratic interpolation done when calculating a wall PDF value. This too is handled by a routine from MATLAB and adds to computation time. Lastly, since the exact position of the boundary is tracked, the computations for setting the boundary position at each node can add to the overhead, especially for a moving boundary. Another potential drawback is that more memory is required for storage. Nodes at the boundary tracking the exact location of the boundary require extra memory to store this information. Memory is also used to store the positions and PDF values for bounce-back and wall particles during one time step so that they don't have to be recalculated for each unknown PDF value sought. This is useful since multiple unknown PDF values (at different nodes) may use some of the same boundary nodes in the multi-dimensional interpolation. Primarily these drawbacks relate to performance issues and ultimately detract from a simplified implementation. However, the upside to the more timely and costly implementation of the MDI treatment is flexibility, a potential increase in accuracy for the lattice Boltzmann method at the boundaries, and simplicity in concept. With advances and progress made in computing technology in the next decade, the drawback of a costly implementation should be reduced in the future.

3.1.3. Particle Dynamics

The particle dynamics implemented in this study will primarily be applied to the moving construct in the RWV bioreactor. The method for handling the particle dynamics in the lattice Boltzmann method for this study combines the concepts seen in literature concerning particle dynamics and the concept behind the MDI boundary treatment. The evaluation of the force on the boundary is slightly modified from the equation proposed by Aidun et al. (1995) using the momentum-exchange method (Equation 2.50). The modification is an attempt at linking the MDI boundary treatment with the force evaluation to maintain an overall consistency in the lattice Boltzmann implementation. The method of Aidun et al. (1998) is used when a moving particle or solid covers or uncovers a fluid node. This involves applying an impulse force to the particle as it covers or uncovers a node. Calculation of the lubrication forces that come about when two surfaces come within one lattice spacing of each other is done using a method similar to that of Nguyen & Ladd (2002). The previous three issues primarily affect the accuracy of evaluating the force applied to the moving construct by the flow. The last issue addressed is the practical solution of the equations of motion for particle dynamics (Equations 2.48 and 2.49). An accurate solution to these equations is important for the accurate modeling of the moving construct.

The equation for calculating the force on a surface using the momentum-exchange method presented by Ladd & Verberg assumes that nodes on both sides of the surface retain fluid (Figure 2.16). This means that nodes that are inside a moving particle still collide and stream PDF values. This is in contrast to the force evaluation method of Aidun et al. (1995) where the nodes inside a solid particle are not treated as having fluid and do not carry out the collision and streaming (Figure 2.17). The method used in this study to evaluate force at a surface is a slightly modified method based on Aidun. Therefore, while fluid will be retained inside the moving construct, it will not be used to model forces. A restriction with the Aidun method is that the boundary is assumed to be halfway between the nodes when calculating the force. Since the MDI boundary treatment already tracks the exact position of the boundary, a modification to the Aidun method is proposed. The following diagram illustrates a boundary that is not halfway between two nodes.

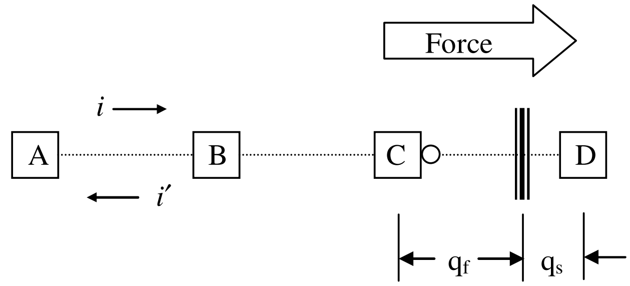


Figure 3.8 – Illustration for Modified Force Evaluation at a Surface

In Figure 3.8, the nodes A, B, and C are in the fluid regime and q_f is the normalized distance from node C to the boundary. Node D is inside the solid particle or in the solid. The normalized distance from the boundary to node D is designated as q_s. Note that $q_f + q_s = 1$. The force of the fluid on the particle is being evaluated. The proposed equation to evaluate the force at halfway through a time step is the following.

$$\mathbf{F}(\mathbf{x}_\mathbf{W}, t + \frac{1}{2}\Delta t) = \frac{\Delta x^3}{\Delta t}\left[2 f_i^c(\mathbf{x}_\mathbf{FLUID}, t) - \frac{2 w_i \rho(\mathbf{u}_\mathbf{W} \cdot \mathbf{e}_\mathbf{i})}{c_s^2}\right]\mathbf{e}_\mathbf{i} \qquad (3.1)$$

Equation 3.1 is almost identical to Equation 2.49, except that the post-collision PDF value of interest is not simply taken as the values at node C. The post-collision PDF value $f_i(\mathbf{x}_\mathbf{FLUID},t)$ is the PDF value associated with a particle in the fluid regime that would stream to the surface in exactly half of a time step (½Δt). The position of this particle is represented by the open circle. For $f_i(\mathbf{x}_\mathbf{FLUID},t)$, the post-collision PDF values $f_i(\mathbf{x}_\mathbf{A},t)$, $f_i(\mathbf{x}_\mathbf{B},t)$, and $f_i(\mathbf{x}_\mathbf{C},t)$ are used to extrapolate this PDF value because $q_f > ½$. Extrapolation is necessary because the particle position lies outside of the known PDF information. This modification in the evaluation of the force adjusts for a boundary that is not exactly halfway in between nodes and has the potential to increase the accuracy of the overall force calculation. This calculation is also more consistent with the MDI boundary treatment because it interpolates PDF values of particles that will hit the wall while the Aidun method was more consistent with the shifted bounce-back boundary treatment which assumes the boundary is halfway between nodes. It is important to note that while the fluid is being retained inside the construct, it is not being used to calculate forces. This is due to the fact that a shell with fluid inside is not the physical model desired. If it is desired, then a simple modification to the Ladd & Verberg equation for force evaluation, Equation 2.44, can be done similar to the one made for the Aidun equation. This modification would account for boundaries that are not exactly halfway between nodes.

For the moving construct, nodes will at times be in the solid regime or fluid regime. Nodes in both regimes will retain fluid, as stated earlier, which makes the transition to different regimes slightly easier in the implementation. All nodes will carry out the collision, streaming, and other processes without having to make exceptions. The method of Aidun et al. will be used in this study for handling nodes that are covered or uncovered by a moving particle. Figure 3.9 shows a moving boundary in the D2Q9 model and can illustrate both cases. In the figure, consider the nodes AIII, AIV, and BIV existing in a solid regime while nodes AI, AII, BI, BII, CI, CII, CIII, and CIV exist in the fluid regime. Node BIII is the node of interest that is either moving from the solid regime to the fluid regime or vice versa.

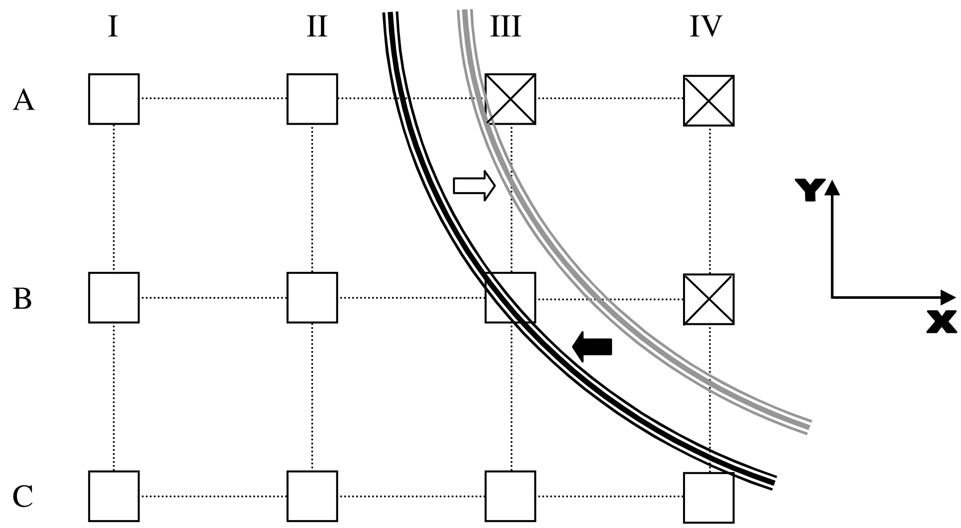


Figure 3.9 – Illustration of a Moving Boundary in the D2Q9 Model

If the boundary is moving from right to left (solid arrow), then node BIII is being covered by the boundary and now exists in the solid regime with nodes AIII, AIV, and BIV. In this case the new density of node BIII will be calculated as the average of the densities at nodes AIII, AIV, and BIV. The more general Equation 3.2 states that the density of a covered node ρ^{CN} is the average density of the closest surrounding N nodes. The velocity of node BIII is set equal to the velocity of the boundary or wall, $\mathbf{u_w}$. This velocity can be calculated from the rigid body motion of the particle. The PDF values are then set using the equilibrium PDF values for a node of with velocity $\mathbf{u_w}$ and density ρ^{CN}. Since a node is covered by the particle, momentum needs to be added to the particle by applying an impulse force. The impulse force and torque that is applied is calculated using Equations 3.3 and 3.4.

$$\rho^{CN} = \frac{1}{N}\sum_{N}\rho_i \tag{3.2}$$

$$\mathbf{F}^{\mathbf{CN}}(\mathbf{x_W}, t+\frac{1}{2}\Delta t) = \sum \frac{\Delta x^3}{\Delta t}\left(2\left(f_i^{\,c}(\mathbf{x_W}-\mathbf{e_i}\Delta t, t) - f_i^{\,CN}(\mathbf{x_W}, t)\right) - \frac{2w_i\rho\left(\mathbf{u_W}\cdot\mathbf{e_i}\right)}{c_s^2}\right)\mathbf{e_i} \tag{3.3}$$

$$\mathbf{T}^{\mathbf{CN}}(\mathbf{x_W}, t+\frac{1}{2}\Delta t) = \left(\mathbf{x_W} - \mathbf{x_{CM}}\right)\times\mathbf{F}^{\mathbf{CN}}(\mathbf{x_W}, t+\frac{1}{2}\Delta t) \tag{3.4}$$

The parameters needed to calculate the impulse force $\mathbf{F}^{\mathbf{CN}}$ are the position of the closest boundary along a link $\mathbf{x_W}$, the post collision PDF values of the nodes neighboring the covered node, and the velocity of the boundary $\mathbf{u_W}$ at $\mathbf{x_W}$. The impulse torque $\mathbf{T}^{\mathbf{CN}}$ is calculated using the force $\mathbf{F}^{\mathbf{CN}}$ and the position of the center of mass of the particle $\mathbf{x_{CM}}$. In Figure 3.9, the uncovering of node BIII occurs when the boundary is moving from left to right (open arrow). The density of an uncovered node is calculated by Equation 3.2 and in this case is the average density of nodes AII, BII, CII, CIII, and CIV. The impulse force and torque are calculated using a slightly modified version of Equations 3.3 and 3.4. The implementation of these rules is virtually identical for the D3Q19 model except that more nodes are involved in the density calculation.

The lubrication force calculations will be handled by a method similar to that of Nguyen & Ladd (2002), where by they calculated the lubrication force between two particles using the following equations. This method was chosen primarily for its simplicity, but also because it can be highly accurate when using well-defined parameters. These equations were previously listed as Equations 2.55 and 2.56.

$$\mathbf{F_l} = -6\pi\mu \frac{a_1^2 a_2^2}{(a_1 + a_2)^2}\left(\frac{1}{h} - \frac{1}{h_N}\right)\mathbf{U_{12}} \cdot \hat{\mathbf{R}}_{\mathbf{12}}, \quad h < h_N \qquad (3.5)$$

$$\mathbf{F_l} = 0, \quad h > h_N \qquad (3.6)$$

The parameter of interest in calculating the lubrication force $\mathbf{F_l}$, will be the cutoff distance h_N. In the study by Nguyen & Ladd, the correct cutoff distance is found for particular particle sizes and fluid viscosities. For this study, a specific cutoff distance h_n will be used based on the analytical lubrication force. Since the distance from a node to a boundary is tracked using the MDI boundary treatment, the distance between particles h, can be easily found. The analytical lubrication force and torque replaces the calculated force between particles if the magnitude of the analytical force is greater than the calculated force. This prevents double counting the force. The following equations are the equations of motion for the particle and must be solved in order to simulate the construct movement in the RWV bioreactor.

$$\mathbf{F}(t) = M\,\frac{d\mathbf{U}(t)}{dt} \qquad (3.7)$$

$$\mathbf{T}(t) = I\,\frac{d\mathbf{\Omega}(t)}{dt} \qquad (3.8)$$

The basic equations of motion for rigid body motion can be solved using a variety of techniques. In this study, these equations are solved using fourth-order Runge-Kutta integration. This integration is done using a routine in the software package MATLAB which also contains the interpolation routine used in the MDI boundary treatment. Further explanation of the specific equations used in conjunction with solving Equations 3.7 and 3.8 is briefly described here. For a more detailed description on physically based modeling, see the notes by Witkin and Baraff (1997).

The state of a rigid body can be described by the following parameters: position, linear momentum, orientation, and angular momentum. The variable representing the state of a rigid body **Y**(t) can be defined as the following in Equation 3.9

$$\mathbf{Y}(t) = \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{P}(t) \\ \mathbf{R}(t) \\ \mathbf{L}(t) \end{pmatrix} \qquad (3.9)$$

The position **x**(t) in Equation 3.9 represents the translation of a rigid body, while the linear momentum **P**(t) is linked to the velocity **U**(t) of the rigid body. The position vector is taken from a fixed point in space, usually denoted as the origin, to the center of mass for the body. The rigid body velocity is related to linear momentum through the mass of the object M, **P**(t) = M · **U**(t). Thus, the derivative of the linear momentum is equal to the force by Equation 3.7.

$$\dot{\mathbf{P}}(t) = \mathbf{F}(t) \qquad (3.10)$$

The orientation **R**(t) in Equation 3.9 represents the rotation of a rigid body, while the angular momentum **L**(t) is related to the angular velocity **Ω**(t) of the rigid body. The orientation is a tensor (not a vector) representing the rotation of the rigid body with respect to some reference orientation. The reference orientation usually is taken with the rigid body naturally aligned with the directions of the major axes. In general, **R**(t) can be represented as the following.

$$\mathbf{R}(t) = \begin{pmatrix} r_{xx} & r_{yx} & r_{zx} \\ r_{xy} & r_{yy} & r_{zy} \\ r_{xz} & r_{yz} & r_{zz} \end{pmatrix} \qquad (3.11)$$

Thus if a point on the rigid body can be described as $\mathbf{p_0}$ with respect to the reference position and orientation (defined as body-space coordinates), then the world-space coordinate of $\mathbf{p_0}$ is **p**(t) = **R**(t)$\mathbf{p_0}$ + **x**(t). In other words, the world-space position of a point on a rigid body can be found using the rotation and the translation information for the body. Figure 3.10 illustrates body-space and world-space. Note here that the rotation of the body can be described by the rotation of the X-Y-Z axes to the X′-Y′-Z′ axes, in the following equation. In Equation 3.12, X′, Y′, and Z′ are column vectors describe the rotation of the original axes.

$$\mathbf{R}(t) = [\mathbf{X}' \quad \mathbf{Y}' \quad \mathbf{Z}'] \qquad (3.12)$$

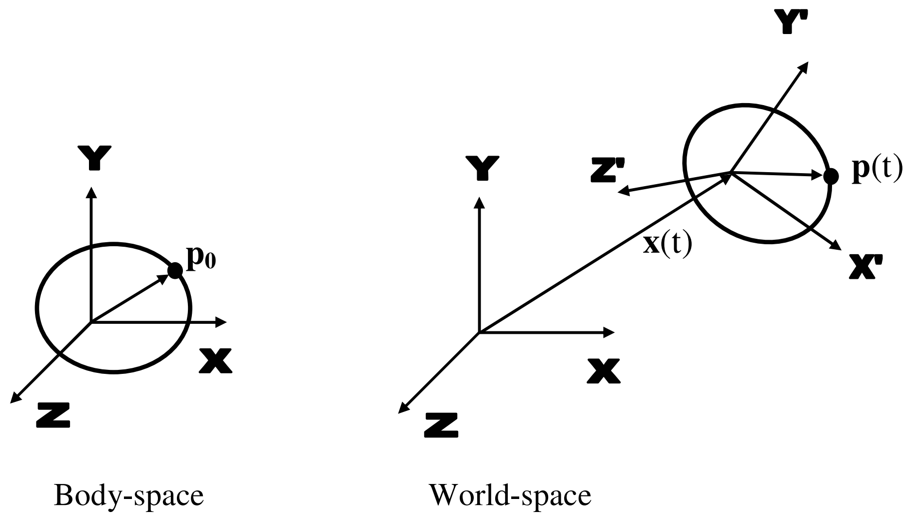


Figure 3.10 – Illustration of Body and World Space Coordinates

In a similar way that position is linked to velocity through its derivative, the orientation is linked to angular velocity through its derivative. By defining a matrix **Ω**(t)* from the angular velocity vector, the derivative of **R**(t) can be expressed as the following.

$$\dot{\mathbf{R}}(t) = \mathbf{\Omega}(t) * \mathbf{R}(t) \qquad (3.13)$$

$$\mathbf{\Omega}(t)* = \begin{pmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{pmatrix} \qquad (3.14)$$

The **Ω**(t)* matrix essentially computes a cross product of **Ω**(t) with each column vector in the **R**(t) matrix. The rigid body angular velocity is also linked to angular momentum through the inertia tensor **I**(t), **Ω**(t) = $\mathbf{I}^{-1}$(t) · **L**(t). Thus, the derivative of angular momentum can be directly related to torque by Equation 3.8.

$$\dot{\mathbf{L}}(t) = \mathbf{T}(t) \qquad (3.15)$$

The inertia tensor **I**(t) of a rigid body is the scaling factor between angular momentum and angular velocity much like mass is the scaling factor between linear momentum and velocity. In general, the inertia tensor describes the distribution of mass within a rigid body and is found by integrating the mass contribution from differential volumes over the entire body. In Equation 3.16, m_i represents the mass of the differential volume and $[x_i, y_i, z_i]$ represents the position of that volume ($\mathbf{p}(t) - \mathbf{x}(t)$) in world-space. The diagonal terms represent the moments of inertia and the off-diagonal terms represent the products of inertia.

$$\mathbf{I}(t) = \begin{pmatrix} \sum m_i \left(y_i^2 + z_i^2\right) & -\sum m_i y_i x_i & -\sum m_i z_i x_i \\ -\sum m_i x_i y_i & \sum m_i \left(x_i^2 + z_i^2\right) & -\sum m_i z_i y_i \\ -\sum m_i x_i z_i & -\sum m_i y_i z_i & \sum m_i \left(x_i^2 + y_i^2\right) \end{pmatrix} = \begin{pmatrix} I_{xx} & I_{yx} & I_{zx} \\ I_{xy} & I_{yy} & I_{zy} \\ I_{xz} & I_{yz} & I_{zz} \end{pmatrix} \tag{3.16}$$

The inertia tensor changes with the orientation of the rigid body, but is easily computed if the inertia tensor in the reference orientation or body-space coordinates is known. For example, consider the rectangular block of uniform unit density with its center of mass located at the origin in body-space coordinates in Figure 3.11. The length, width, and height of the block is x_0, y_0, and z_0 respectively. The inertia tensor with the block naturally aligned with the axes, defined as $\mathbf{I_{body}}$, is easily calculated in Equation 3.17.

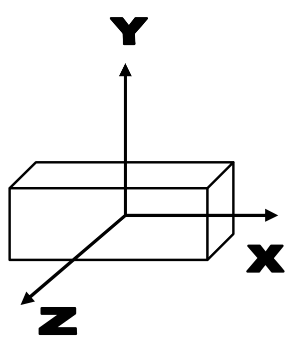


Figure 3.11 – Illustration for $\mathbf{I_{body}}$ Calculation

$$\mathbf{I_{body}} = \frac{M}{12}\begin{pmatrix} y_0^2 + z_0^2 & 0 & 0 \\ 0 & x_0^2 + z_0^2 & 0 \\ 0 & 0 & x_0^2 + y_0^2 \end{pmatrix} \qquad (3.17)$$

$$\mathbf{I}(t) = \mathbf{R}(t)\mathbf{I_{body}}\mathbf{R^T}(t) \qquad (3.18)$$

Since the axes are aligned with the rectangular block in Figure 3.11, the moments of inertia for the block are easily calculated and lie on the diagonal of the $\mathbf{I_{body}}$ tensor in Equation 3.17. Note that there are no products of inertia in this case. Once $\mathbf{I_{body}}$ is known the I(t) at any time can be calculated using Equation 3.18. This equation requires the orientation of the rigid body **R**(t), and the transpose of the orientation $\mathbf{R^T}$(t). Now, using the information in Equations 3.10, 3.13, and 3.15 the derivative of the state vector **Y**(t) can be expressed as the following.

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} \mathbf{x}(t) \\ \mathbf{P}(t) \\ \mathbf{R}(t) \\ \mathbf{L}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{U}(t) \\ \mathbf{F}(t) \\ \mathbf{\Omega}(t) * \mathbf{R}(t) \\ \mathbf{T}(t) \end{pmatrix} \qquad (3.19)$$

The difficulty in solving Equation 3.19 stems from the fact that the orientation is a tensor and not a vector. Since the orientation is a tensor, numerical drift will become a factor and build up in the coefficients of **R**(t). Another way to represent the orientation is by using quaternions rather than rotation matrices. The advantage of doing so is that a quaternion is a four element vector that is less susceptible to numerical drift. For further information on quaternions and the related mathematics in using quaternions see Appendix A. With the use of quaternions, where **q**(t) represents a quaternion, Equation 3.19 becomes Equation 3.20.

$$\frac{d}{dt}\mathbf{Y}(t) = \frac{d}{dt}\begin{pmatrix} \mathbf{x}(t) \\ \mathbf{P}(t) \\ \mathbf{q}(t) \\ \mathbf{L}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{U}(t) \\ \mathbf{F}(t) \\ \frac{1}{2}\mathbf{\Omega}(t)\mathbf{q}(t) \\ \mathbf{T}(t) \end{pmatrix} \qquad (3.20)$$

Equation 3.20 represents the set of equations that can be used to solve the motion of a rigid body. Again, in this study the solution for the motion of the construct is found using a fourth-order Runge-Kuttta method to integrate Equation 3.20. The solution at each time step requires the initial position, linear momentum, orientation, angular momentum, force, and torque. The force and torque is found using the momentum-exchange method at each half time step (t – ½Δt and t + ½Δt), but then averaged to find the force and torque at time t. The force and torque are then assumed to be constant throughout that time step. The solution at the end of the time step will give the final state variable **Y**(t) for the construct. This final state then serves as the initial condition for the next time step. From the state variable, the velocity and angular velocity can be calculated using the following equations.

$$\mathbf{U}(t) = \frac{\mathbf{P}(t)}{M} \qquad (3.21)$$

$$\mathbf{\Omega}(t) = \mathbf{I}^{-1}(t)\mathbf{L}(t) \qquad (3.22)$$

The accuracy of the methods discussed for the particle dynamics in the lattice Boltzmann simulation will be verified through several numerical studies. These studies will include verifying the accuracy of the overall dynamic behavior of a construct and the forces calculated with momentum-exchange method. The specific aims are to reproduce the dynamics of a construct with accuracy, test whether the interpolation boundary treatments can improve the accuracy of the momentum-exchange method, and show that the interpolation boundary treatments can robustly handle moving boundaries.

3.1.4. Grid Refinement

There have been several methods proposed to utilize a non-uniform grid in the lattice Boltzmann method. The method chosen for this study is a blend of the concepts proposed by He et al. (1996), Filippova & Hänel (1998a, 1998b, 2000), and Lin & Lai (2000). The underlying concept in the grid refinement method for this study is the same concept on which the MDI boundary treatment is based. The idea of sufficiently smooth surfaces for PDF values proposed by He et al. is taken into consideration because it links traits of the lattice Boltzmann model with application. The scaling concept of PDF values between grids by Filippova & Hänel is also necessary in a correct grid refinement implantation. Scaling ensures that similarity of flow characteristics is achieved on different grids. Lastly, a second-order interpolation on the sub-grid boundary of incoming PDF values as done by Lin & Lai should preserve accuracy on the boundary of the primary grid and sub-grid. These important concepts taken from previous literature are all incorporated in the grid refinement method used in this study. The grid refinement method presented will primarily focus on refinement of primary grid, but also has the capability of coarsening the primary grid.

The coarsening of the primary grid is achieved through a similar method previously described by He et al. Nodes can be placed at distances apart other than the lattice spacing Δx. The exact post-advection position of streamed particles can be tracked and the PDF values needed at node sites can be reconstructed using information in multiple dimensions. Consider Figure 3.12 where nodes are located on a non-uniform grid in the D2Q9 model. Nodes in column I and II are spaced at a distance of Δx, nodes in column II and IV are spaced at a distance of 2Δx, and there are no nodes in column III. Particles with the $\mathbf{e_5}$ velocity that stream from nodes in column IV will end up in column III. Their post-advection positions are denoted by the solid circles. Particles streaming from row II will end up precisely at column I and their positions are noted by the checkered circles.

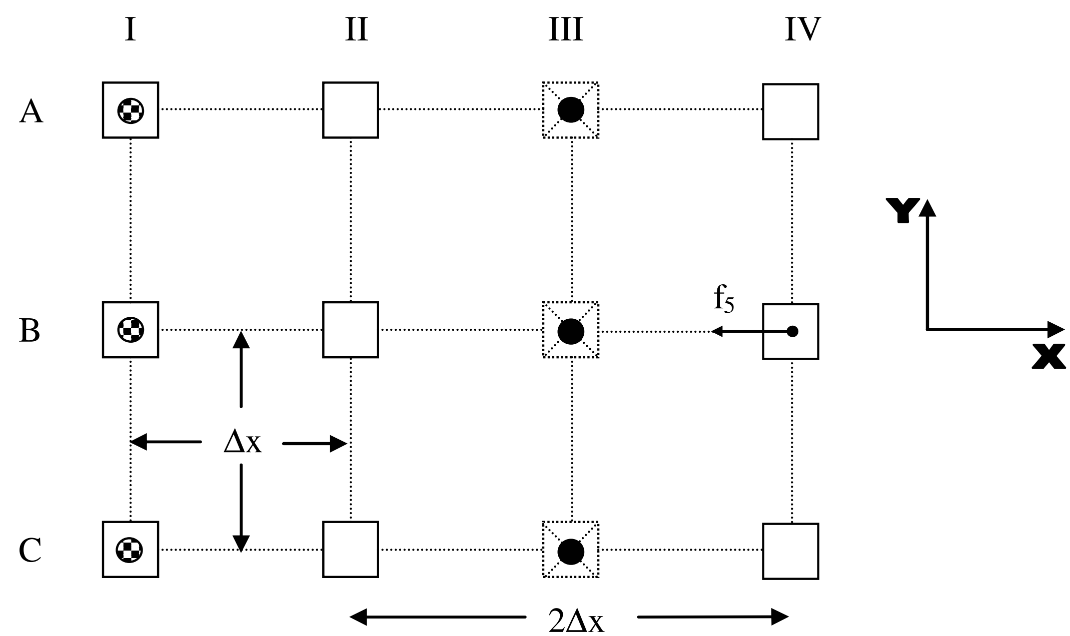


Figure 3.12 – Illustration for a Non-Uniform Coarsened Grid

In this case, the nodes in column II do not have a post-advection value at the end of the time step. For example, the PDF value $f_5(\mathbf{x_{BII}},t+\Delta t)$ needs to be found. In this study, the value would be found through an interpolation similar to the one used in the MDI boundary treatment. A surface is fitted to the PDF information already known in the surrounding space: $f_5(\mathbf{x_{AI}},t+\Delta t)$, $f_5(\mathbf{x_{BI}},t+\Delta t)$, $f_5(\mathbf{x_{CI}},t+\Delta t)$, $f_5(\mathbf{x_{AIII}},t+\Delta t)$, $f_5(\mathbf{x_{BIII}},t+\Delta t)$, and $f_5(\mathbf{x_{CIII}},t+\Delta t)$. The PDF value at node BII is then found from the fitted surface. While some of the streamed particles ended up at exactly at their neighboring nodes, this is not necessarily a requirement. As long as there is enough PDF information to faithfully reconstruct part of the sufficiently smooth surface, a PDF value can be reconstructed. This not only aids in coarsening a grid in this study, but can be instrumental to forming more complex non-uniform grids.

The refinement of the grid can be accomplished using several techniques presented in literature. For this study, refinement is done by overlaying a sub-grid on top of the main grid. Figure 3.13 illustrates the sub-grid existing on top of a main grid for the D2Q9 model. The sub-grid refines the main grid by a factor of N and will execute the lattice Boltzmann method. This means that for every time step on the main grid, the sub-grid will execute N time steps (Figure 2.20). In Figure 3.13, the sub-grid interfaces with the main grid along column III.

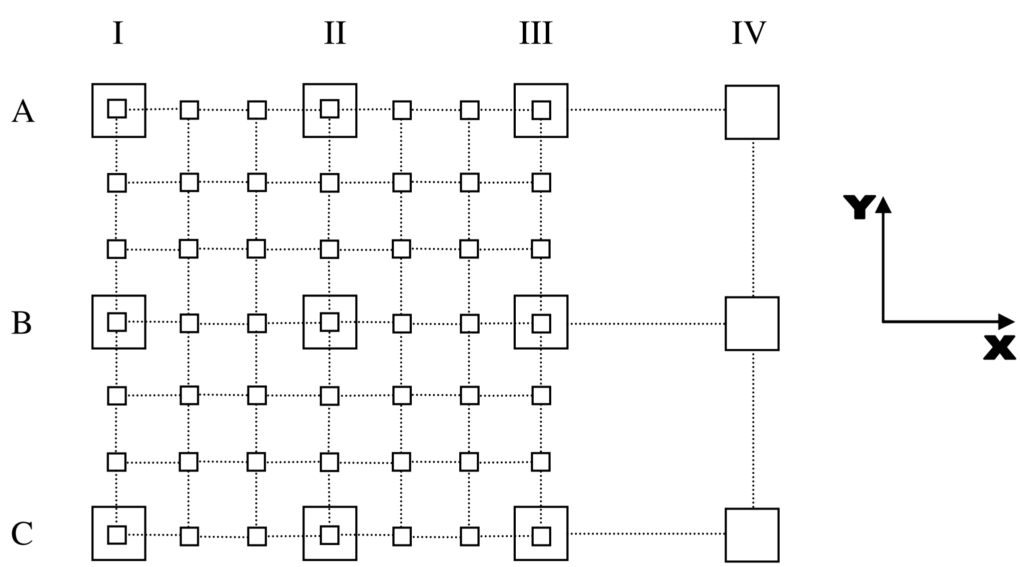


Figure 3.13 – Illustration of a Sub-Grid with N = 3 in the D2Q9 Model

The initialization of the sub-grid PDF values occurs by interpolation. Initial PDF values on the sub-grid are found by fitting an interpolation surface to PDF value from the main grid. In this case, PDF values at nodes AI, AII, AIII, BI, BII, BIII, CI, CII, and CIII are used. Information from surrounding nodes on the main grid is also used to increase interpolation accuracy at the edges of the sub-grid. Interpolation is done for all velocities in the model used. For the D2Q9 model, this interpolation is done nine times corresponding to the nine velocities in the model. Once these values are in place, the lattice Boltzmann method can commence on the sub-grid.

The PDF values at the boundaries of the sub-grid are needed for each sub-grid time step. In general, these values are interpolated from the main grid. In a previous study, the PDF values used in the interpolation are first scaled according to the method of Filippova & Hänel (Equation 2.58). This scaling is not performed for this study. However, note that the relaxation constant on the sub-grid must be transformed (Equation 2.57) in order to maintain similar flow characteristics between the two grids. The PDF values from the main grid are directly used to interpolate boundary PDF values on the sub-grid. For Figure 3.13, PDF values at nodes AIII, BIII, and CIII are used to interpolate PDF values along column III for the sub-grid. This occurs at every sub-grid time step in order to provide PDF values at the boundary that would have streamed from the main grid space. In this study, quadratic interpolation is done in space, but linear interpolation is done in time. For example, consider the interpolation of a PDF value associated with the $\mathbf{e}_5$ velocity in Figure 3.13 and the timeline in Figure 3.14.

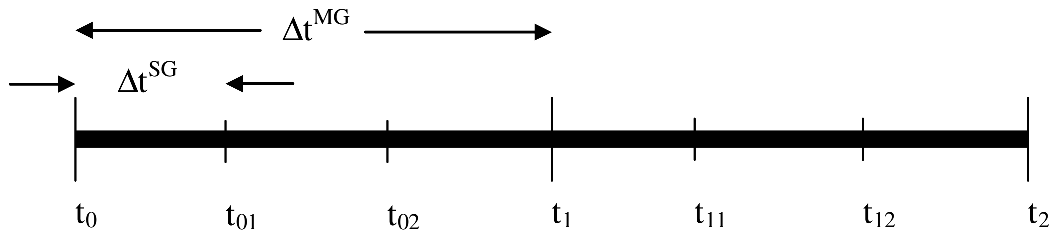


Figure 3.14 – Timeline for Sub-Grid with N=3

Assuming the sub-grid is initialized at $t = t_0$, the lattice Boltzmann method will first proceed on the main grid to $t = t_1$. Once this is complete, the lattice Boltzmann method on the sub-grid will commence and reach $t = t_{01}$. Before completing this time step and moving on, the PDF values on the boundary of the sub-grid are filled using interpolation. For a PDF value $f_5(\mathbf{x_{SGB}},t_{01})$ along column III, a quadratic interpolation using the scaled values $f_5(\mathbf{x_{AIII}},t_0)$, $f_5(\mathbf{x_{BIII}},t_0)$, and $f_5(\mathbf{x_{CIII}},t_0)$ is done first. This yields a value $f_5(\mathbf{x_{SGB}},t_0)$. Next another quadratic interpolation using scaled values at the next main grid time step is done using the scaled values $f_5(\mathbf{x_{AIII}},t_1)$, $f_5(\mathbf{x_{BIII}},t_1)$, and $f_5(\mathbf{x_{CIII}},t_1)$. This yields a value $f_5(\mathbf{x_{SGB}},t_1)$. Finally, a linear interpolation in time is done to find the needed boundary value $f_5(\mathbf{x_{SGB}},t_{01})$. This method is similar to Lin & Lai (2000) in that higher order interpolation is done at the boundaries. The method is extended to 3D by fitting a higher order surface to the spatial information, but still interpolating linearly in time.

This method diverges from some of the proposed grid refinement techniques in literature because information is not passed back to the main grid. Filippova & Hänel (1998a) proposed scaling PDF values (Equation 2.59) that would stream into the main grid at the sub-grid boundary. Yu et al. (2002) proposed a method by which the sub-grid replaces the main grid rather than just overlapping. There is some built-in overlap of nodes at the boundary, where scaling and interpolation operations occur in order to fully integrate the sub-grid into the main grid. This type of method is not pursued in this study because it limits the flexibility of the grid refinement. The method proposed in this study, allows both grids to coexist simultaneously. A sub-grid can be placed or removed on top of the main grid at any time in course of the main grid's execution of the lattice Boltzmann method. The sole purpose of the sub-grid is to refine the flow in key areas of interest. Because of this flexibility, this study proposes that grid refinement can be used for a moving construct such that the sub-grid tracks with the motion of the construct. In other words, a sub-grid can follow the motion of a construct by removing and placing itself as the construct moves. The primary concerns with this implementation are how accurate the method can be with a shifting sub-grid and how much time (on a sub-grid scale) is required for the shifting sub-grid to attain its best accuracy.

Numerical studies will test the effectiveness of the grid coarsening and refinement methods. These methods are based on the sufficiently smooth surface postulation and rely on interpolation by fitting surfaces to PDF information. The numerical studies will primarily focus on examining the flow surrounding the construct in the RWV bioreactor because this flow is of primary interest. Grid refinement should allow for better estimates of the shear stresses felt by the constructs. The flexibility of the method also should allow a better resolution of the flow around a moving construct in the RWV bioreactor. However, the robustness and accuracy of these methods needs to be verified.

3.1.5. Shear Stresses

The evaluation of the shear stresses felt by the construct is of primary interest in this study. In most of the previous studies done with the lattice Boltzmann method, the calculation of shear stresses has not been a point of focus. For the current study, the viscous stress tensor for a Newtonian fluid will be calculated. In general the viscous stress tensor for an incompressible fluid can be described by the following equations (Batchelor 1967).

$$\boldsymbol{\sigma}_{\mathbf{ij}} = -P\delta_{ij} + 2\mu \mathbf{e}_{\mathbf{ij}} \qquad (3.23)$$

$$\mathbf{e}_{\mathbf{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \qquad (3.24)$$

In Equation 3.23, $\boldsymbol{\sigma}_{\mathbf{ij}}$ is the stress tensor, P is the static-fluid pressure, δ_{ij} is the Kronecker delta, and μ is the dynamic viscosity. The rate-of-strain tensor, $\mathbf{e}_{\mathbf{ij}}$, also appears in Equation 3.23 and is defined in Equation 3.24. The diagonal elements of $\boldsymbol{\sigma}_{\mathbf{ij}}$ (σ_{xx}, σ_{yy}, and σ_{zz}) represent the normal stresses and the off diagonal elements (σ_{zy}, σ_{yz}, and σ_{zx}) are the shearing stresses. For a Newtonian fluid, the strain rate is linearly related to stress through the dynamic viscosity as seen in Equation 3.23. These equations along with the shear rate (Equation 3.32) are the basis of the analysis performed.

In this study, the shear stresses in the fluid within the RWV bioreactor will be calculated using Equations 3.23 and 3.24, but using two different techniques. The velocity partial derivatives will be calculated by using velocities and straight-forward finite difference formulas or by using the PDF information within the lattice Boltzmann method. As an example, for a 2D calculation the normal stress σ_{xx} and the shearing stress σ_{xy} can be calculated using the following equations.

$$\sigma_{xx} = -P + 2\mu\frac{\partial u}{\partial x} \qquad (3.25)$$

$$\sigma_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \qquad (3.26)$$

The static-fluid pressure can be calculated using Equation 2.7. The dynamic viscosity can be calculated from the density (Equation 2.5) and kinematic viscosity (Equation 2.9). The velocity partial derivatives like $\partial u/\partial x$ and $\partial v/\partial x$ can be calculated by using finite difference formulas. Equations 3.27, 3.28, and 3.29 are three options in calculating $\partial u/\partial x$.

$$\frac{\partial u}{\partial x} = \frac{u(\mathbf{x}+\mathbf{e_1}\mathit{\Delta} x,t) - u(\mathbf{x}-\mathbf{e_1}\mathit{\Delta} x,t)}{2\Delta x} + O(h^2) \tag{3.27}$$

$$\frac{\partial u}{\partial x} = \frac{-3u(\mathbf{x},t) + 4u(\mathbf{x}+\mathbf{e_1}\mathit{\Delta} x,t) - u(\mathbf{x}+2\mathbf{e_1}\mathit{\Delta} x,t)}{2\Delta x} + O(h^2) \tag{3.28}$$

$$\frac{\partial u}{\partial x} = \frac{u(\mathbf{x}+\mathbf{e_1}\mathit{\Delta} x,t) - u(\mathbf{x},t)}{\Delta x} + O(h) \tag{3.29}$$

If there are no boundaries present, a central difference formula can be used as in Equation 3.27. Equations 3.28 and 3.29 are forward difference formulas that can be used if a boundary is present at **x-Δx**. Backward difference formulas are used if a boundary is present at **x+Δx** and take on the same general form as Equations 3.28 and 3.29. The three-point formula in Equation 3.28 is always used over Equation 3.29 as long as the data exists for all three points. With these simple finite difference formulas, the calculation of the stress tensor can be easily accomplished.

The velocity partial derivatives in Equations 3.25 and 3.26 can also be found by taking the derivatives of the particle distribution functions (PDFs) that make up the calculation of the u velocity component (Equation 2.6). For example, $\partial u/\partial x$ and $\partial v/\partial x$ can be calculated by Equations 3.27 and 3.28 in the D2Q9 model (Figure 2.6).

$$\frac{\partial u}{\partial x} = \frac{\partial}{\partial x}(u) = \frac{\partial}{\partial x}\left(c\frac{f_1 + f_2 + f_8 - f_4 - f_5 - f_6}{\rho} \right) \tag{3.30}$$

$$\frac{\partial v}{\partial x} = \frac{\partial}{\partial x}(v) = \frac{\partial}{\partial x}\left(c\frac{f_2 + f_3 + f_4 - f_6 - f_7 - f_8}{\rho} \right) \tag{3.31}$$

Assuming that the PDFs are smooth and differentiable in space, the further breakdown of the derivatives within these equations is straightforward. An example of the final calculation for these partial derivatives is provided in Appendix B. The final calculations include several partial derivatives of particle distribution functions. These are calculated using finite-difference formulas similar to those presented earlier in Equations 3.27 to 3.29 except that instead of the u velocity component, a PDF value would be used. This second method in calculating the velocity partial derivatives not only provides a check for the straight finite difference method, but also lends credibility to the smooth and differentiable assumption for the PDFs.

The last calculation of interest related to the stress tensor involves the strain rate calculation. The strain rate, also referred to as shear rate, within a fluid has units [1/s] and is defined as S. Equation 3.32 is the definition of S^2 (FLUENT V6.1 User's Guide).

$$\begin{aligned} S^2 = &\left[\frac{\partial u}{\partial x}\left(\frac{\partial u}{\partial x}+\frac{\partial u}{\partial x}\right)+\frac{\partial u}{\partial y}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)+\frac{\partial u}{\partial z}\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)\right]+ \\ &\left[\frac{\partial v}{\partial x}\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right)+\frac{\partial v}{\partial y}\left(\frac{\partial v}{\partial y}+\frac{\partial v}{\partial y}\right)+\frac{\partial v}{\partial z}\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)\right]+ \\ &\left[\frac{\partial w}{\partial x}\left(\frac{\partial w}{\partial x}+\frac{\partial u}{\partial z}\right)+\frac{\partial w}{\partial y}\left(\frac{\partial w}{\partial y}+\frac{\partial v}{\partial z}\right)+\frac{\partial w}{\partial z}\left(\frac{\partial w}{\partial z}+\frac{\partial w}{\partial z}\right)\right] \end{aligned} \tag{3.32}$$

The shear rate parameter can give an indication of shear stresses felt by tissue constructs. An example is given by Neitzel et al. (1998). In this study, maximum shear rate values corresponded to shear stresses on the order of 0.8 dynes/cm^2. This parameter is automatically calculated in the CFD package FLUENT and can be used to estimate shear stresses within the fluid. It will be used as a point of comparison for the lattice Boltzmann software and FLUENT in verifying the lattice Boltzmann implementation.

3.1.6. Implementation Issues

There are several other implementation issues concerning the lattice Boltzmann method that should be mentioned. Most of these issues are practical matters that are rarely mentioned in literature, but are important in understanding the implementation of the method overall. For this study, some of these issues include types of geometries, placement of boundaries, application of a body force, implementation of flow inlets and outlets, and convergence criteria.

There are several geometries used in this study. Flow in channels, cavity driven flow, and flow in an annulus are the primary cases of interest. All involve the use of a square or cubic lattice depending on whether 2D or 3D flow is being studied. In general, the overall shape of the main grid used for all cases will be rectangular and depends on the resolution chosen. The boundaries at the edges of the main grid are assumed to be at a distance of one-half lattice spacing away. The boundaries that determine the geometry of the case are then placed within the rectangular main grid. For example, consider Figure 3.15 which illustrates the grid layout for the case of channel flow in the D2Q9 model.

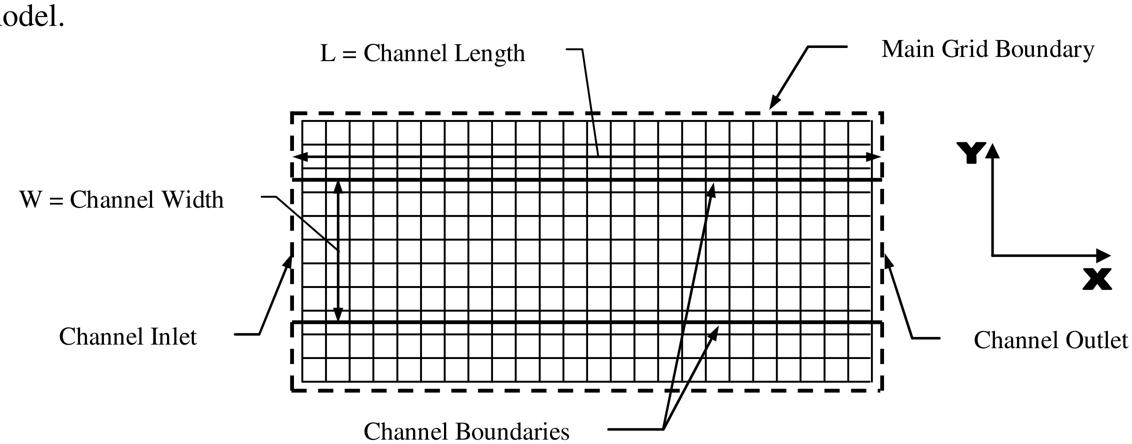


Figure 3.15 – Illustration of Grid Layout for Channel Flow in the D2Q9 Model

In this case, the resolution of the channel flow is set as 6 nodes across by 25 nodes long. The main grid is then chosen to be 12 nodes by 25 nodes and a physical boundary is assumed around the edges. Choosing the main grid slightly larger than the desired resolution of the channel flow allows for flow calculations on both sides of the channel boundaries. While this extra space results in more calculations for flow outside of the area of interest, it allows for a consistent approach that facilitates other geometries and other aspects such as moving boundaries. In Figure 3.15, the dashed lines represent the physical boundaries for the main grid and the channel. For this case, it is assumed that the channel boundaries are placed halfway between the nodes. Thus the total channel length L equals $25\Delta x$ and the total channel width W equals $6\Delta x$. For simulations in this study, the lattice spacing Δx is given a physical dimension. In this case if Δx is chosen as 0.1m, then W equals 0.6m and the L equals 2.5m. The time step Δt for this grid would then be chosen as 0.1s.

Boundary placement is done by storing the location of the boundary along each link of a node. In this case, the nodes along the channel boundary will store the exact position of the boundary. For example, nodes that are in the flow and border the top channel boundary store information telling that along the links associated with the $\mathbf{e}_2$, $\mathbf{e}_3$, and $\mathbf{e}_4$ velocities a boundary is $0.5\Delta x$ away. For boundaries aligned with the lattice, implementation is straight-forward. However, for the placement of curved boundaries a root-finding technique is used to find the exact position of a boundary along a link. Consequently, if a boundary is known to exist along a link, the exact position of the boundary is the solution to an equation and for this study that root is found using the bisection method. Figure 3.16 is an illustration of the grid layout for the 2D annulus case where curved boundaries are part of the geometry of the problem.

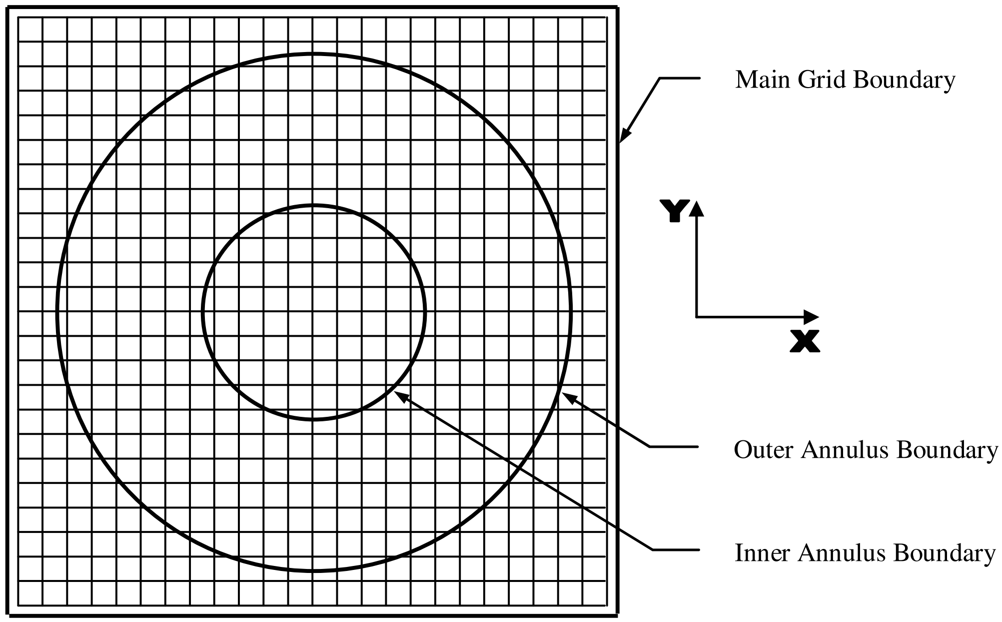


Figure 3.16 – Illustration of Grid Layout for 2D Annulus Case

The relaxation parameter τ can be chosen to obtain a desired kinematic viscosity. If the desired kinematic viscosity υ is 0.001m^2/s, then the relaxation parameter can be calculated using Equation 3.33.

$$\tau = \frac{1}{2}\left(6\upsilon\frac{\Delta t}{\Delta x^2} + 1\right) \qquad (3.33)$$

Using Equation 3.23, which is simply Equation 2.9 rearranged, τ is calculated as 0.53 for the 2D channel case. Similar to the lattice spacing, the mean flow density ρ_m is also given a physical dimension. For this case, the mean density is chosen as 1.0kg/m^3. This value needs to be specified for several calculations including those for pressure and forces. In addition, the initial PDF values at each node are calculated using mean density in the EPDF equation and taking the node velocity equal to zero. Note that the PDF and EPDF values are in units of density.

The channel flow can be driven in one of four ways: a body force can be applied to drive the flow, a translating wall can be implemented, a pressure gradient can be imposed, or a velocity profile can be enforced at the inlet. In order to apply a body force, a discrete amount of density is added to and subtracted from PDF values at each node. This density ρ_{BF} is calculated using an Equations 3.34 and 3.35, which were previously presented by Zou et al. (1995) and He et al. (1995).

$$PG = 8\frac{\rho \upsilon}{W^2}u_{\max} \qquad (3.34)$$

$$\rho_{BF} = PG\frac{\Delta x}{c^2} = PG\frac{\Delta t^2}{\Delta x} \qquad (3.35)$$

In Equation 3.34, the value PG represents the pressure gradient required to drive a 2D Poiseuille flow in a channel of width W such that the maximum centerline velocity is u_{max}. If the required u_{max} in this case is 0.01m/s, the necessary pressure gradient PG is calculated as 0.00022N/m^3. The required density value to drive the flow ρ_{BF} is then calculated in Equation 3.35 as 0.000022kg/m^3. The application of the body force in practice involves adding ρ_{BF} to the PDF value associated with the velocity direction of the flow. In other words, for the D2Q9 model and Figure 3.15, if the flow is moving from left to right, then at all nodes in the flow ½ ρ_{BF} would be added to f_1 and subtracted from f_5. Only ½ρ_{BF} is added and subtracted because the overall density is unchanged. This calculation is expressed in Equations 3.36 and 3.37.

$$f_1 = f_1 + \frac{1}{2}\rho_{BF} \qquad (3.36)$$

$$f_5 = f_5 - \frac{1}{2}\rho_{BF} \qquad (3.37)$$

This application of the body force is done after collision but before streaming. The result of applying the body force is 2D Poiseuille flow. The last issue that needs to be addressed with the body force application is the inlet and outlet conditions. Typically when a body force is used to drive the flow, the inlet and outlet conditions are set as periodic conditions. For example in this case, particles at the channel outlet that stream with the $\mathbf{e_1}$ velocity will re-enter the flow domain at the channel inlet. In the same fashion, particles at the channel inlet that stream with the $\mathbf{e_5}$ velocity will re-enter the flow at the outlet.

Driving the flow with a moving wall is accomplished by assigning the channel boundaries a velocity. The application of momentum-exchange rules, which are described in Equations 2.42 and 2.43, drives the flow. If only one boundary is assigned a velocity, then the boundary is essentially translating and 2D Couette flow is the result. The inlet and outlet conditions are typically set as periodic conditions for this case.

The last two cases are flows driven by a pressure gradient and flows driven by specifying an inlet velocity. Both of these cases can be accomplished by applying a boundary treatment to the inlet and outlet nodes. In 2D, the hydrodynamic and non-equilibrium bounce-back boundary treatments can be applied to implement a specified inlet velocity and pressure gradient, respectively. For applying a pressure gradient, the density at the inlet and outlet can be specified while solving for PDF values that stream into the flow in the non-equilibrium bounce-back boundary treatment. The density difference in order to create the correct pressure gradient ρ_{PG} can be calculated using the following equations.

$$\rho_{PG} = PG \frac{L}{c_s^2} \qquad (3.38)$$

$$\rho_{inlet} = \rho_m + \frac{1}{2}\rho_{PG} \qquad (3.39)$$

$$\rho_{outlet} = \rho_m - \frac{1}{2}\rho_{PG} \qquad (3.40)$$

The value for ρ_{PG} in this case using a channel length L of 2.5m in Equation 3.38 is 0.00165kg/m^3. Therefore using Equations 3.39 and 3.40 the inlet and outlet densities calculated, ρ_{inlet} and ρ_{outlet}, are 1.000825kg/m^3 and 0.999175kg/m^3 respectively. These values are used in conjunction with equations similar to Equations 2.24, 2.25, and 2.26 to calculate PDF values for particles that stream into the channel at the inlet and outlet. The previous equations presented (2.24 – 2.26) are not exactly the equations needed because the non-equilibrium bounce-back boundary treatment depends on the orientation of the boundary.

For the case of a specified inlet velocity, equations similar to Equations 2.13 – 2.16 for the hydrodynamic boundary treatment can be used. A parabolic velocity profile can be calculated and serve as the known values for u and v in these equations for the inlet of the channel. The solution of the equations yields the density and the PDF values for particles that stream into the flow. Like the non-equilibrium bounce-back boundary treatment, the exact equations needed for a particular situation depends on the orientation of the boundary. Again, this serves as a hindrance in implementation because the equations are geometry dependent and the treatments are difficult to extend to 3D. The outlet of the channel can be handled as a free outlet. This can be accomplished by implementing the extrapolation boundary treatment. The PDF values associated with particles that stream into the channel at the outlet can be extrapolated from known PDF values. This is demonstrated in Figure 2.12 and Equation 2.35. In contrast to the previous boundary treatments mentioned, this method for a free outlet can be easily extended to 3D and does not depend on boundary orientation. The disadvantage is that the treatment may suffer from inaccuracies due to the use of extrapolation.

Alternative boundary conditions can be applied in the case of a channel where flow is driven by the sedimentation of a particle. Periodic conditions can be applied at the inlet and outlet, however this simulates a periodic array of particles falling together. The shifted bounce-back boundary treatment can be applied to the inlet and outlet, effectively setting the velocity to zero and creating a closed cavity. This approximates more real world conditions as real channels are finite. Lastly, the PDF values that stream into the flow at the inlet and outlet can be set to the equilibrium PDF values of a node initialized by the average density. This is termed as the free boundary condition or Stokes boundary condition because at low Reynolds numbers (Stokes flow), the fluid density measured at the inlet and outlet better approximates that of a fluid in a longer or infinite channel. These boundary conditions are used for the numerical studies involving particle sedimentation.

Once the boundary treatments, boundary conditions, or body force that drives the flow are implemented, the lattice Boltzmann method continues to step in time until convergence is observed. Convergence is observed for steady flow solutions and is determined by the criteria set by the user. The convergence criterion for a steady flow used in this study is based on the L_2 norm of the errors associated with velocity.

$$\frac{\sum_{i,j} \left\| \mathbf{u}(\mathbf{x_{i,j}}, t+\Delta t) - \mathbf{u}(\mathbf{x_{i,j}}, t) \right\|}{\sum_{i,j} \left\| \mathbf{u}(\mathbf{x_{i,j}}, t) \right\|} \leq 1.0x10^{-6} \qquad (3.41)$$

This criterion is the same as the criterion proposed by He et. al (1996). In Equation 3.41, the L_2 norm is denoted by $\| \cdot \|$ and is taken for the error of velocity in the numerator. In this study, this criterion is useful for the steady flows that will be simulated in order to evaluate the accuracy of the lattice Boltzmann method. Typically for flows that are driven by a body force or by a moving wall, this criterion is met and exceeded. However, in the case of pressure driven or velocity driven channel flows, the criterion may be met but is usually not exceeded. The criterion is less useful for the transient flow inside the RWV bioreactor with a moving construct as this flow is unlikely to converge.

3.2. Programming Software

For the past several decades, high level programming has been done in languages like FORTRAN, C, or C++. The goal of high level programming languages is to minimize the effort needed to program and promote hardware independence. FORTRAN was originally developed in the 1950's as the first high level language by IBM and lends itself to scientific and engineering applications. Traditionally, FORTRAN has been favored by the scientific and engineering community because of its ability to efficiently process mathematical equations. C was developed in the 1970's at Bell labs and is more closely related to systems programming and the development of operating systems like UNIX and Linux. Modularity is a strength of C that aids in portability and execution speed. C++ was developed in the 1980's at Bell labs and was intended as a multi-paradigm or hybrid language. Object-oriented programming is one of the paradigms of C++ and it emphasizes predefined classes and functions which can be associated with programming objects. Other features of object-oriented programming like inheritance, class libraries, and polymorphism contribute to easier software development. C++ also has the flexibility to compile and run code written in C if the programmer requires it. In searching for the programming software to use in this study, several considerations were made. The languages mentioned (FORTRAN, C, and C++) each have their strengths and weaknesses. Ultimately, C++ was chosen for this study because of its potential to ease the development process with object-oriented programming.

Object-oriented programming in contrast to procedural programming (the paradigm of C and FORTRAN) allows flexible functionality to be built into objects that will interact in order to accomplish the goals of a program. In developing an object-oriented program, the programmer can think about a problem in real-world terms (assigning objects attributes and functions and relationships) rather than in procedural terms. Using FORTRAN, although very suitable for scientific applications, may have taken a significantly longer time to develop a working program from scratch with the same flexibility. To address the possible weakness of C++ in scientific computing, the software package MATLAB is also used in this study to augment the capabilities of C++. MATLAB allows for the use of pre-written mathematical routines that assist in complex computations. Without the use of MATLAB, development time would also be extended. The software development environment that allowed for object-oriented programming and had distinct features that would assist in shortening development time was Visual C++.

3.2.1. Visual C++ 6.0

The software development environment Visual C++ 6.0 (VC++) is used in this study. VC++ was developed by Microsoft (Redmond, WA – www.microsoft.com) and is primarily used in software development for the Windows operating system. VC++ has all the characteristics of a typical C++ development environment with a few exceptions. One distinct feature is that VC++ incorporates the use of a library called Microsoft Foundation Classes (MFC). MFC is a library that provides a class hierarchy that is useful for programming and development in Windows. Many of the typical Microsoft products like Word and Excel are programmed in VC++ and specifically make use of MFC. In this study, MFC provides a framework onto which the lattice Boltzmann software is built. The only disadvantage of developing in MFC is that the software is limited to Windows machines only. The software can be ported to work on UNIX or Linux based machines, but this would take time and effort. However, this disadvantage is somewhat minimized by the growing use of Windows machines for computational work.

Another advantageous feature of VC++ is the ease of development of graphical user interfaces (GUIs) that are used in the software. A graphical user interface is not often a priority for CFD applications, especially in those developed by academic researchers, but has been found very useful in this study. The GUI for the lattice Boltzmann software has allowed the program to be more flexible than a traditional procedural program developed in C or FORTRAN given the same time frame and experience needed for development. While a GUI can be programmed in many languages, the framework provided by VC++ allows one to be built quite easily. Figure 3.17 illustrates the GUI for the lattice Boltzmann software developed in this study.

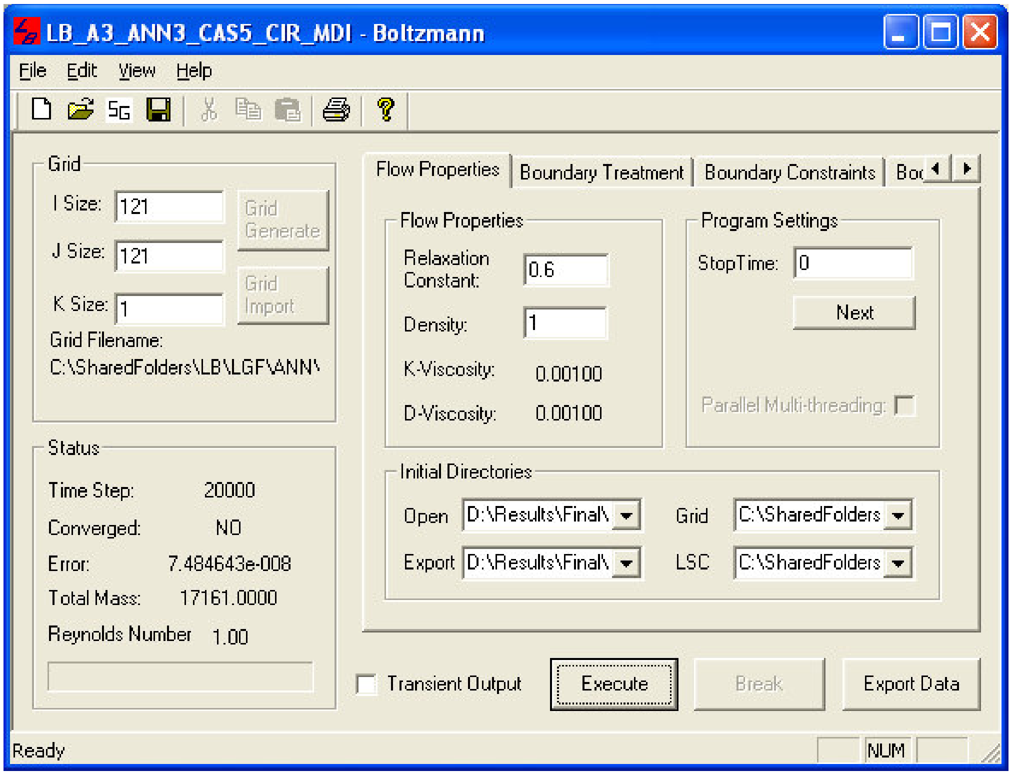


Figure 3.17 – GUI for the Lattice Boltzmann Software

The lattice Boltzmann software is based on a single-document interface (SDI) model much like the Microsoft programs Notepad and Paint. This means that GUI will display and run only one document (or simulation in this case) at a time. In the case of Figure 3.17, the simulation entitled "LB_A3_ANN3_CAS5_CIR_MDI" is being run. A multiple-document interface (MDI) model (which programs like Word are based on) was not chosen for this study because a single-document approach was a better fit and less complex.

From the GUI developed for this software many functions of the program are accessible. These functions deal with the simulation parameters themselves. Grid geometry can be specified or previously used grids can be loaded. Flow properties such as relaxation parameter and mean density can be easily set. Also, the type of boundary treatment can be specified as well as how the flow is to be driven (moving wall, pressure gradient, etc.). The GUI also allows for the changing of parameters in the middle of a simulation if it is required. For example, a construct can be changed from a stationary construct to a moving construct midway through a simulation. Other GUI functions deal with handling the information associated simulations. For example, the status of the simulation is updated on the GUI. Basic bookkeeping functions such as saving or loading a simulation and exporting data to a readable format are also accomplished through the GUI. The software can automatically export data at desired time steps for transient analysis as well. Overall, the flexibility and ease in running many kinds of simulations stems from the GUI.

VC++ may not be the optimum development environment for CFD applications, but it does have a lot to offer for academic researchers who may not be very experienced in computer science or programming. Many of these features that afford the flexibility of the program can be had using other languages or developer environments, but VC++ allows these features to be accessible to novice programmers in a short amount of time.

3.2.2. MATLAB

In this study, MATLAB Release 12 is used to supplement VC++. MATLAB is a software package developed by The Mathworks (Natick, MA – www.mathworks.com) and is primarily used for technical computing and modeling. Specifically, the C/C++ Math library and MATLAB Complier are used in this study in order to augment the functionality and simplify the development of the lattice Boltzmann software. The

C/C++ Math library allows a programmer using C or C++ to have access to most of the mathematical functions available in MATLAB. Programming in MATLAB primarily deals with matrix operations, thus typical functions used in MATLAB involve manipulating matrices and solving linear systems. The use of these functions involves adding the library to your software and some MATLAB tailored programming in C++ to accommodate the use of the library. The MATLAB Compiler allows the compilation of MATLAB script files (.m extension files) into C or C++ code. The compiler is useful if writing code for particular sections in MATLAB is easier or preferred. Both of these tools are used in this study.

The functions used from the C/C++ library involve the interpolation needed for the MDI boundary treatment. Specifically, the functions *griddata* and *griddata3* allow for the fitting of an interpolation surface to input data. These functions are the workhorses behind the MDI boundary treatment. The difference between the two is that *griddata* fits a 3D surface to PDF values in the 2D lattice Boltzmann model and *griddata3* fits hyper-surfaces to PDF values in the 3D lattice Boltzmann model. The following illustrations in Figures 3.18 and 3.19 are a visualization done in MATLAB of the MDI boundary treatment along a curved boundary for the D2Q9 model. The interpolation is for a PDF value of a particle with the $\mathbf{e}_1$ velocity. Figure 3.18 is a top view of the 3D surface fitted by MATLAB. The curved boundary is on the left of the surface. The open circles represent streamed, wall, and bounce-back particle positions and the small x represents the position where interpolation is needed. Figure 3.19 is an isometric view of the 3D surface and gives a better feel for how the surface looks in 3D space.

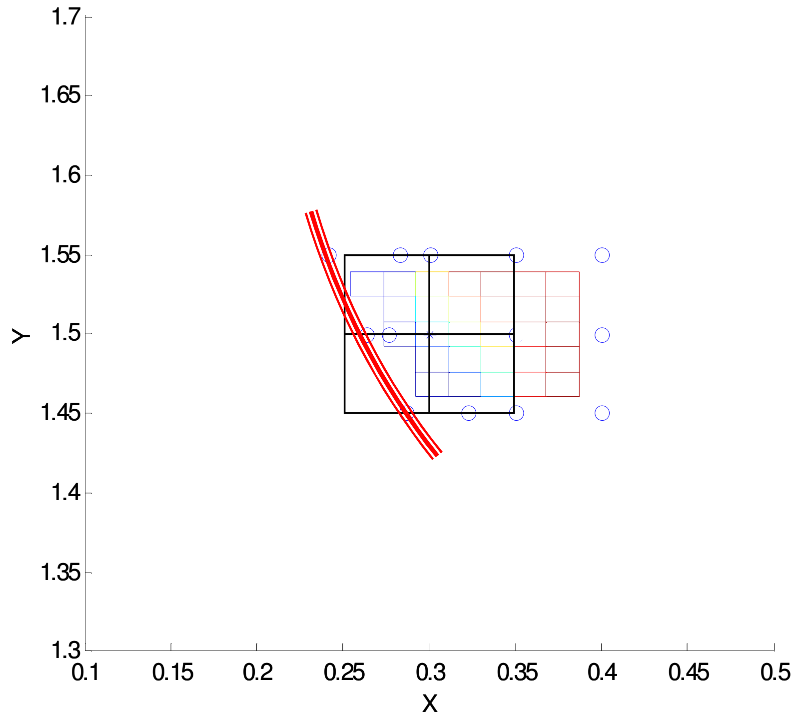


Figure 3.18 – Illustration of the MATLAB Surface Fitting with griddata (top view)

The primary use of the MATLAB Complier in this study is for code related to the particle dynamics. The MATLAB function *ODE45* solves a system of differential equations using a higher-order Runge-Kutta method. This function isn't readily available in the C/C++ math library so the code for a small section of the lattice Boltzmann code was written in MATLAB and complied into C++ code. The code involved the solution of Equation 3.20 which describes the motion of a particle. The convenience of having complex functions available directly through the C/C++ Math library or indirectly through the MATLAB Complier was invaluable in the development of the code. The libraries may not have the most efficient or optimized code for C++, but that is precisely the tradeoff in using pre-packaged code.

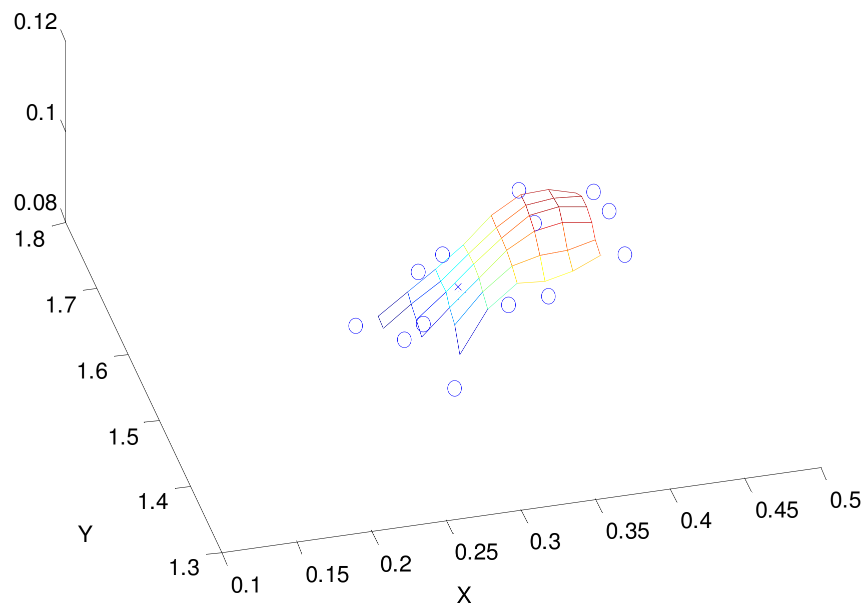


Figure 3.19 – Illustration of MATLAB Surface Fitting with griddata (isometric view)

While the MATLAB Compiler and the C/C++ Math library are an integral part of the implementation of the lattice Boltzmann software, the MATLAB software itself is also used in this study. MATLAB is a powerful tool for computations and visualization and is used in the post-processing of data from the lattice Boltzmann software. Specifically, comparisons of computational data using different methods are done in MATLAB and are aided by its vast library of functions. For more details, the reader can refer to The Mathworks website or the MATLAB technical reference manual.

3.3. Parallel Implementation

There have been several published studies by researchers who have implemented the lattice Boltzmann method on parallel systems. The lattice Boltzmann method lends itself to parallelization because of its local explicit computations. Their primary focus of these studies has been more on the efficiency of the method on parallel systems versus the development of the method. Their interests were how much faster the code runs when parallelized (efficiency or speedup) and how parallel code can be optimized to run faster (parallelization strategies or domain decomposition). Studies by Ujita et al. (1998), Satofuka & Nishioka (1999), and Kandhai et al. (1998) have explored the efficiency of lattice Boltzmann simulations on parallel systems. Ujita et al. did a number of 2D simulations including a Karman vortex street, bent piping flow, and flow in porous media. They found that parallelization efficiency was high for their lattice Boltzmann simulations when compared to finite difference schemes. Satofuka & Nishioka simulated the 2D and 3D decay of homogeneous isotropic turbulence. They examined the effects of domain decomposition of the efficiency of a parallelized lattice Boltzmann code. They found they could minimize CPU time through a specific decomposition and observed that speedup was higher in 3D. Kandhai et al. examined flow through porous medium and explored different parallelization strategies. They concluded that higher efficiencies can be achieved in the lattice Boltzmann using a decomposition method that effectively load-balances work to individual computation nodes.

While these studies are insightful, they were mostly done on large supercomputers with multiple processors having shared or distributed memory. The code was most likely written in a high-level language like FORTRAN or C and parallelization was done using the Message Passing Interface (MPI) library. This situation is not applicable to the current study because of software limitations. However, in this study an attempt to parallelize the code is made. Strategies for parallelization or the efficiency or speedup with a parallel program will not be explored in depth. One reason for this is that software designed to run on individual Windows machines typically is not meant for parallelization. Another reason is that the primary focus of this study is not parallelization of the lattice Boltzmann method, but developing and testing the method itself. Software that allows for networked Windows based machines to run parallelized code and act as one computer includes PVM and NT-MPICH.

3.3.1. PVM

PVM stands for Parallel Virtual Machine and was developed through the research of heterogeneous network computing. Originally developed at Oak Ridge National Laboratory in 1989 by Sunderam and Geist, PVM is an integrated set of software tools and libraries that emulates a general-purpose, flexible, heterogeneous concurrent computing framework on interconnected computers of varied architecture (Geist et al. 1994). PVM version 3 was released in 1993 and was used in high-performance scientific applications around the world. PVM transparently handles all tasks associated with parallel network computing on computers with different architectures. While the original PVM software worked on UNIX based machines, a PVM port to Windows NT was done by Fischer and allowed networked WIN32 machines to be included (Geist et al. 1994). The software is not being actively supported today as some of the original PVM researchers have moved onto a new project involving network computing called Harness. The lattice Boltzmann software was first parallelized with PVM because of the ease in initial implementation. However, since PVM is not being actively supported, the software NT-MPICH, which uses the more widely accepted standard MPI library, was also explored.

The programming paradigms that PVM supports relating to process structure are crowd and tree type computations. In one type of crowd computation, a master node or the node running the control program can initiate or spawn processes on slave nodes to perform computations on separate parts of the workload. Visually, this relationship between a master and slaves looks more star-like. In a tree computation, processes can be spawned in a tree-like manner where by the relationships look more like parent-child. Child nodes can become parents by spawning more nodes. Figure 3.20 illustrates these two types of computation paradigms.

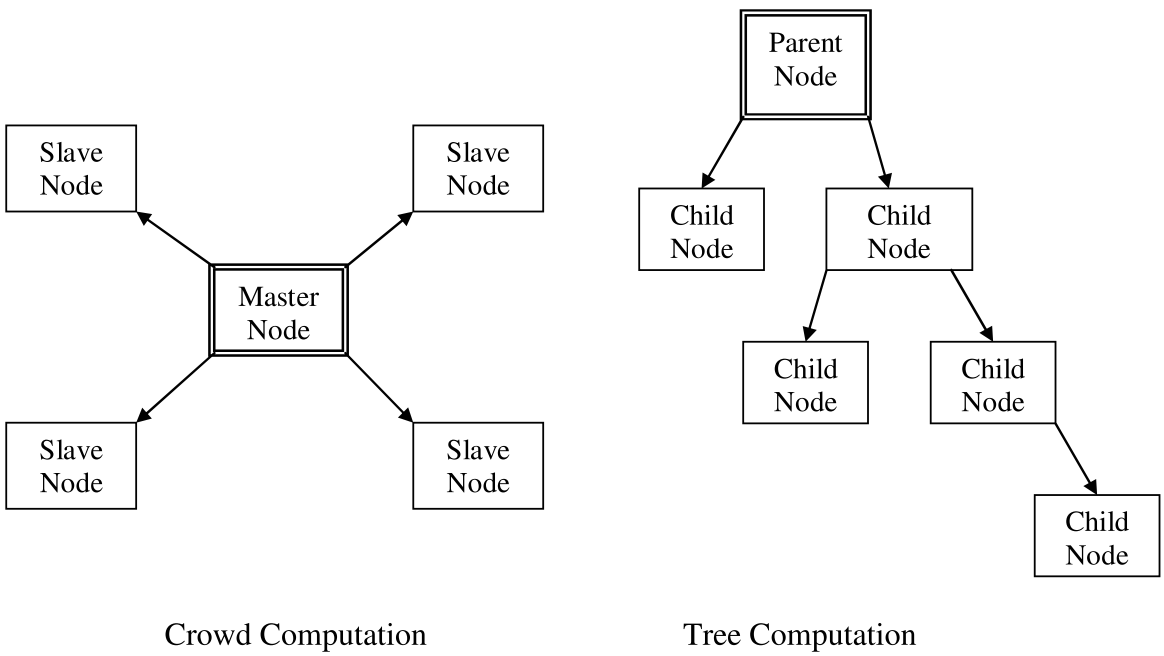


Figure 3.20 – Two Common Parallel Computation Paradigms

The choice of computation paradigm depends on the application. If the total workload is known before hand, a master-slave type paradigm may be sufficient. If the total workload is not known when the program is started, then a tree computation may be more efficient. PVM supports both types and also supports a hybridization of these two types. That means a program can spawn nodes in both a master-slave and parent-child type relationship. More information on general parallel computation paradigms can be found in literature. The PVM implementation of the lattice Boltzmann software is based on a master-slave type paradigm because in general, the workload can be assessed beforehand.

The main lattice Boltzmann software is started on a master node and in general will take care of most of the lattice Boltzmann functionality. If the software is running in parallel, the master node will spawn other processes on slave nodes or other computers networked to the master node. In practice, the PVM daemon must be first started in order to open communications between the master node and the slave nodes. This is done on the master node. Next, the lattice Boltzmann software can be started on the master Windows workstation and number of available slave nodes or hosts can be displayed. When enabled, the PVM code within the lattice Boltzmann software then starts a slave process on each available node. The slave processes contain only a portion of code related to the computations necessary for the lattice Boltzmann. The slave nodes wait for messages to be passed to them from the master node that contain data for processing. Once the slave node completes the computation, it passes a message back to the master node with the result.

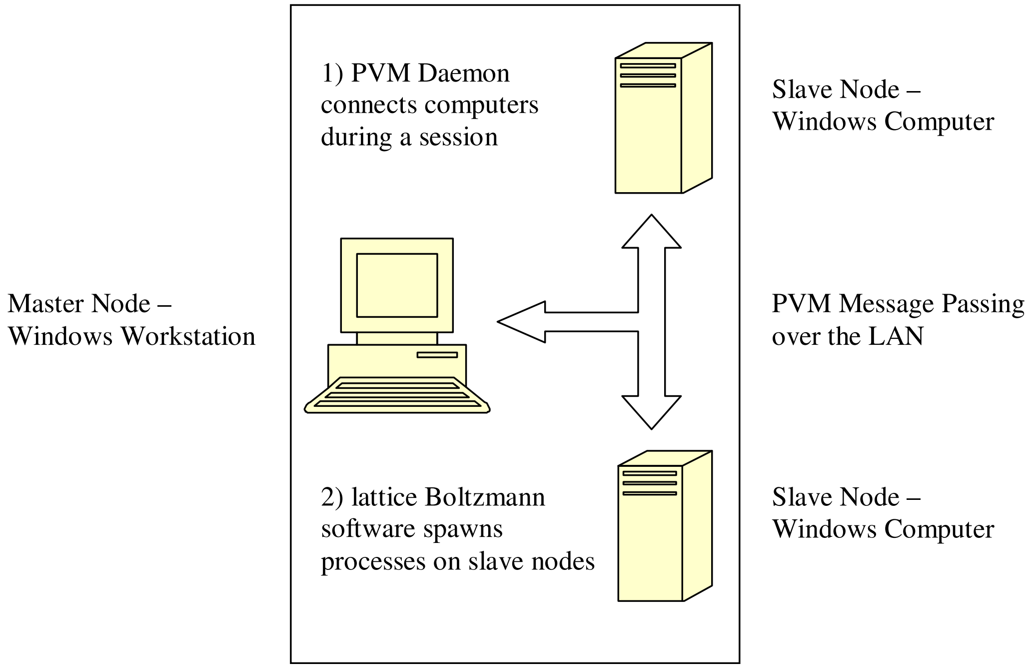


Figure 3.21 – PVM Implementation of Lattice Boltzmann Software

In the lattice Boltzmann software, an example of a typical bottleneck for computations is the interpolation functions related to the MDI boundary treatment. In the PVM implementation, these interpolation functions are handled by slave nodes and results are relayed back to the master node. This interaction between the master and slave nodes is illustrated in Figure 3.21. Ideally, most of the repetitive functions done in the lattice Boltzmann method can be parallelize to speedup the program execution.

The PVM implementation of the lattice Boltzmann software functions adequately. While only a portion of the code is parallelized, there is an improvement in speed of the parallel implementation over the serial implementation of the code. However because PVM is not well supported and somewhat difficult to maintain, optimizing the code has not been pursued in PVM. An NT-MPICH parallel implementation is being explored because the MPI library is an accepted standard that will be supported and developed into the future.

3.3.2. NT-MPICH

The Message Passing Interface (MPI) library was developed by a community of vendors, implementers, and users. MPI defines a standard message-passing system that was built to run on massive multi-processor parallel supercomputers and on workstation clusters alike. The main advantage of having a standard is portability. Programs written in FORTRAN, C, and C++ can implant the MPI library. There are several implementations of MPI available including those that are vendor supplied and some free implementation. MPICH is a free implementation that was developed by Argonne National Laboratory and Mississippi State University. It is based on an earlier portable message passing language called Chameleon. MPICH allows for the basic implementation of MPI programs on the Windows platform. The software MP-MPICH stands for Multi-Platform MPICH and was developed by Aachen University. MP-MPICH is also freely available like MPICH and was developed before MPICH had added Windows support. Within MP-MPICH is the Windows port of MPICH called NT-MPICH which has the ability to run on all UNIX platforms and on Windows machines. NT-MPICH differentiates itself with MPICH in that it uses a different communications device which provides better performance. NT-MPICH also offers a tool to start programs on remote computers called RexecShell. In this study, NT-MPICH is the MPI implementation used for the lattice Boltzmann software.

The computational paradigms supported by MPI are different from those used in PVM. The paradigms that MPI supports are related more to workload allocation rather than process structure. In NT-MPICH, programs are typically started either through the command line tool MPIEXEC or using the RexecShell GUI tool. This process involves either a single program (meaning the same program is launched on all processors) or multiple programs (different programs are launched on different processors). In a basic structured MPI program, unlike in PVM the program itself does not spawn another program or process when needed. The number of processes started is determined by the user by way of command line parameters or GUI tool configuration. Thus, the paradigms that MPI supports pertain to a single or multiple programs working on different data (SPMD or MIMD). These are types of data parallelism. Thus the program may not have a "master" program in the SPMD paradigm. The master may be considered the program that is running on the primary user's node, but typically the data workload is split evenly between all programs. The other type of workload allocation may involve functional parallelism. This is where different functions of a program are being performed in parallel, while the data is not decomposed or split to be worked on in parallel. While PVM supports functional parallelism, the paradigm doesn't suit MPI quite as well.

The lattice Boltzmann software has also been written to support NT-MPICH. The implementation of the software involves first starting up the program through RexecShell, the GUI tool provided in NT-MPICH. Through RexecShell, the user starts the program on all nodes or hosts specified by the user configuration simultaneously. In order for this to work, NT-MPICH must be installed on all computers and the background application rclumad.exe must be running. In order to check the connection, the lattice Boltzmann software has a system test function to test connectivity and the RexecShell GUI also has I/O windows for the programs as well. Once the lattice Boltzmann software starts executing, it will pass messages to the other processes on other nodes. The other nodes typically handle the MDI interpolation functionality exactly like the PVM implementation. Figure 3.22 illustrates the NT-MPICH implementation of the lattice Boltzmann software.

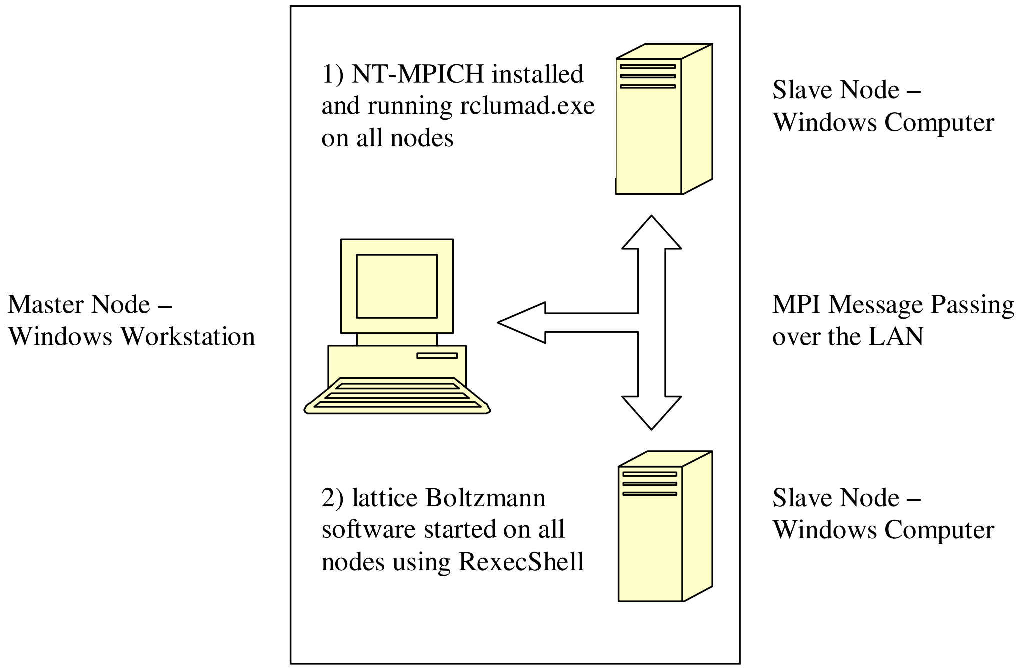


Figure 3.22 – NT-MPICH Implementation of Lattice Boltzmann Software

The major differences between the PVM implementation and the NT-MPICH implementation include how the message passing software is configured before running the software and how the software actually runs. For PVM, a session must be started whereby the PVM Daemon connects the computers to open communications. Once this is done, the lattice Boltzmann software can be started on the master node. In NT-MPICH, the computers are always ready after installation to open communications as long as rclumad.exe is running on all the computers. The software must be started using either the command line or the RexecShell tool in order to open communications. The other major difference is that in PVM, slave processes are spawned by the master node while in NT-MPICH the processes are all started together. While there are other small differences between PVM and NT-MPICH, it seems they both have advantages and disadvantages. Performance-wise in initial testing they are comparable. Implementation-wise they both have desirable features. However, computations for the numerical studies will be done using NT-MPICH because MPI has the advantage of being more portable. In addition, the MPI standard has much more user acceptance and support in the long term.

3.4. Summary

This chapter details the current implementation of the lattice Boltzmann method. The intent in the design of this implementation was to incorporate flexibility by using the simplest lattice Boltzmann techniques and models and user-friendly programming software. This was done so that further advancements in the lattice Boltzmann method could be added to this model easily. Basic models for boundary treatments, grid refinement, and particle dynamics are implemented and will be tested for their accuracy in both 2D and 3D. In the future, the use of lattice Boltzmann models for heat and mass transfer can also be built upon the current methods. The programming software used is also flexible so that improvements can be done easily in future versions of the code. Thus, the implementation used is intended to be universal in application and versatile into the future.

Chapter 4

Numerical Studies, Evaluation, and Results

The process by which results from a CFD code is given credibility to base design decisions is termed verification and validation. An NPARC website based on CFD verification and validation (Slater) gives a thorough introduction to verification and validation. The American Institute of Aeronautics and Astronautics (AIAA) defines verification and validation as the following:

> Verification – The process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model. (AIAA G-077-1998)
>
> Validation – The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model. (AIAA G-077-1998)

The difference between verification and validation is that verification determines if the implementation of a conceptual model is correct while validation determines if a computational simulation agrees with physical reality (Slater). The process by which a code is verified includes checking the code for programming bugs, examining grid convergence, and comparing solutions to highly accurate results. The process by which a code is validated includes examining grid convergence, comparing solutions to experimental data, and examining model uncertainties. For further detail on the guidelines for verification and validation, the interested reader can refer to the AIAA publication (AIAA G-077-1998) or the reference by Roache (1998). The present study attempts to follow in part these guidelines for verification and validation of the lattice Boltzmann code.

To verify the accuracy of this implementation of the lattice Boltzmann method, several numerical studies were performed and evaluated. Initially, the numerical studies test implementation methods in 2D and 3D involving boundary treatments, grid refinement, and dynamics. Further studies attempt to simulate the dynamic RWV bioreactor environment. Evaluation of the results of these studies will validate the usefulness of the lattice Boltzmann methods and will assist the comparison of the simulation results with experimental results obtained in previous research. This chapter details the numerical studies that were performed and the evaluation techniques that were used in analyzing the results. The chapter then describes the results of the studies and references their illustrations in Appendices C and D.

4.1 Numerical Studies

The numerical studies presented involve a mixture of simple benchmarking problems to more complex dynamic simulations. Each numerical study presented tests and helps evaluate specific portions of the lattice Boltzmann implementation. The three main areas of numerical studies test the boundary treatment, grid refinement, and dynamics. Boundary treatment studies aim to test the accuracy of the MDI treatment compared with other previously used boundary treatments. The grid refinement studies test the accuracy of the current grid refinement implementation and help evaluate if grid refinement adequately resolves areas of interest in the flow domain. The force and dynamics studies test the accuracy of force evaluation in the lattice Boltzmann method by examining different implementation methods. These studies also calculate the shear stress and strain rates within the flow field for comparison. The final set of numerical studies presented will include the simulation of the floating construct in the RWV bioreactor environment. These numerical studies help thoroughly assess the quality of the lattice Boltzmann implementation so that valid insights and conclusions can be drawn.

4.1.1 Boundary Treatment Studies

The primary purpose of the boundary treatment studies is to test the accuracy of the MDI boundary treatment implementation and compare its accuracy to previously used boundary treatments. The importance of boundary treatments stems from the fact that their performance directly affects the accuracy of the solution. Researchers have previously proposed boundary treatments that are second-order accurate in space. This implies that the error term diminishes fast with respect to the characteristic grid scale. The numerical studies here should assess whether the MDI treatment is second-order accurate. The studies also evaluate the unqualified accuracy of the solutions produced with various boundary treatments. There were primarily three geometries that are used to test the boundary treatments: flow in a channel, lid-driven flow in a cavity, and flow in an annulus.

Poiseuille flow with shifted boundaries

Poiseuille flow is generated using a body force in a 2D channel. Body force driven flow in a 3D square channel also is examined. The channel boundaries are placed at varying distances from the computation nodes in order to shift the boundaries. Figure 4.1 illustrates the 2D shifted channel with the normalized boundary distance Δ and the channel width of L. The maximum flow velocity is set as U_1. The variation of boundary distances tests the accuracy of the boundary treatments as the results are compared with the analytical solution. In addition, the number of nodes across the channel is increased in order to examine the behavior of the error as the grid is refined.

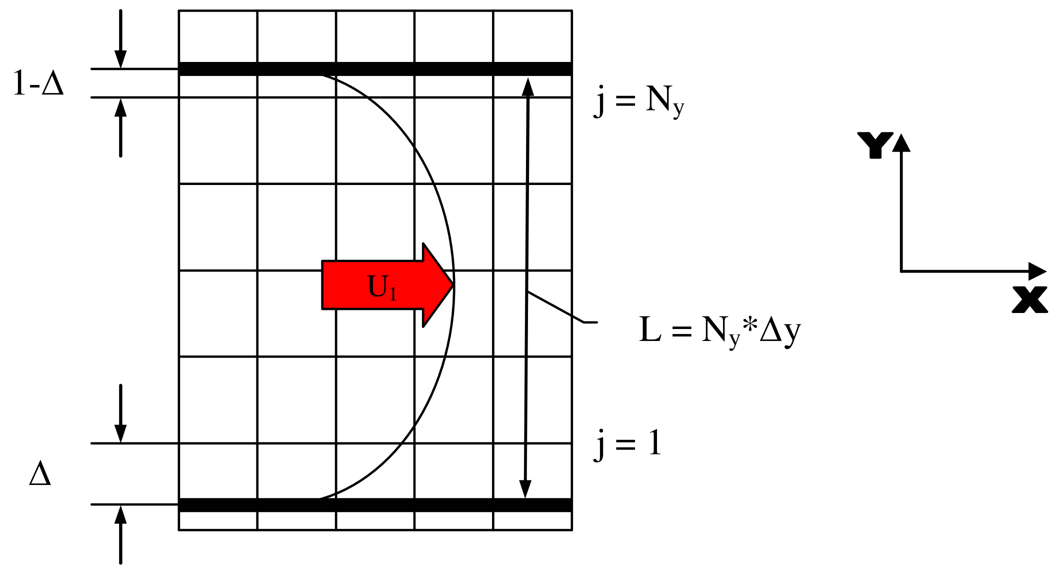


Figure 4.1 – 2D Channel Flow Geometry

In a similar 2D study done by Verberg & Ladd (2000) using the continuous bounce-back boundary treatment showed that the largest deviation of velocity at the smallest channel resolution (4 computation nodes across the channel) from the analytic solution was 10%. Also, the mean relative error decreased with $1/N_y^2$, which indicated second-order error convergence. In a similar study done by Yu et al. (2003), the wall slip velocity and relative L_2-norm error for Poiseuille flow in a 2D channel was examined. The distance of the boundary was varied and the number of nodes increased in order to examine the error. Second-order convergence of the wall slip velocity as well as the relative error was observed.

For this study, the channel always has a fixed lattice spacing Δy with a varying resolution of nodes across the channel. This means that N_y and L will change for the fixed Δy. Error behavior in relation to L is examined. In addition, the normalized boundary distance Δ is varied between 0.5 and 1 for each resolution in order to create the shifting channel. The different boundary treatments that are tested include the MDI treatment, the Bouzidi interpolation treatment, and the Yu linear treatment. The shifted bounce-back treatment is used when $\Delta = 0.5$, while the other treatments are used for $\Delta \neq 0.5$. The maximum velocity deviation from the analytical solution as well as the RMS error for a profile across the channel is calculated using equations similar to Equations 2.63 and 2.64.

In the 3D case, body force driven flow in a square channel is examined. The same general procedure for analysis is used. The lattice spacing, Δx, will be fixed for all resolutions. The normalized boundary distance Δ will be varied between 0.5 and 1. This boundary distance will be used for two adjacent sides of the rectangular channel, while the opposite sides have a boundary at a distance $1 - \Delta$ away. The maximum velocity is set and denoted as U_1. A cross section of the rectangular channel is shown in Figure 4.2. The variation of the boundary distance Δ creates the shifting channel. RMS error will be examined for the lattice Boltzmann results when they are compared to the analytical solution of pressure driven flow in a rectangular channel. As in the 2D case, error behavior in relation to L also is examined. The analytical solution, taken from Rosenhead (1963), is shown in Equations 4.1 – 4.4 and is derived from the general rectangular solution. In this case it is specific to the square channel geometry.

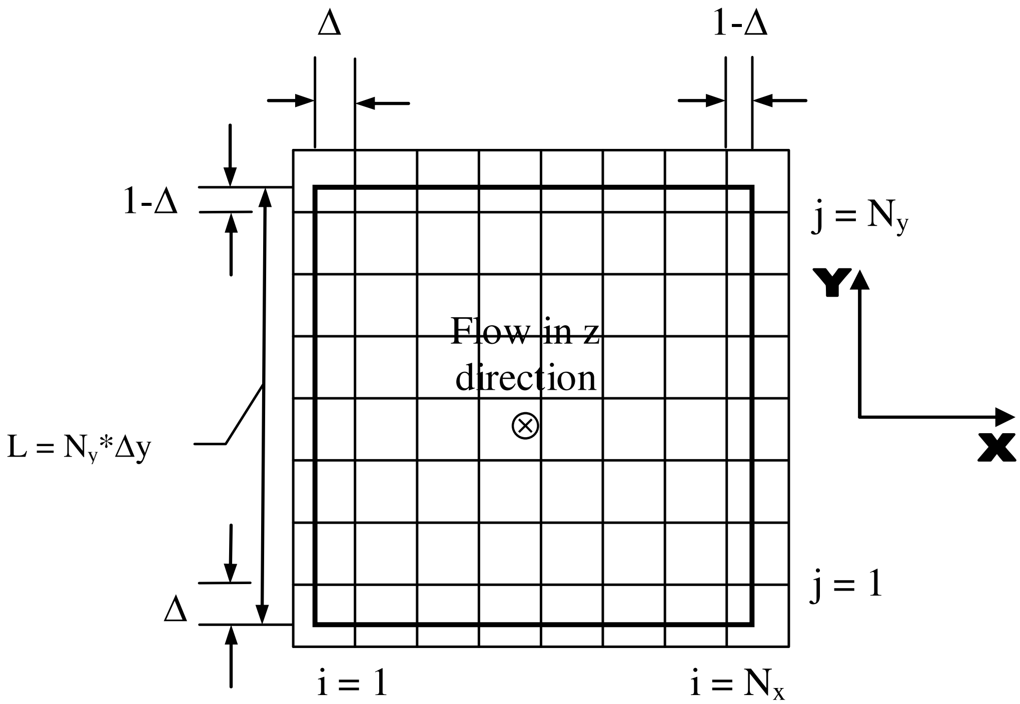


Figure 4.2 – Cross Section of Shifted Channel for 3D Flow

$$w^* = \frac{1}{2}\left(1\text{-}y^{*2}\right)\text{-}2\sum_{n=1}^{N}\frac{(\text{-}1)^n}{\alpha_n^3}\cos\left(\alpha_n y^*\right)\frac{\left(\exp\left[\alpha_n(x^*-1)\right]+\exp\left[-\alpha_n(x^*+1)\right]\right)}{\left(1+\exp^{-2\alpha_n}\right)} \tag{4.1}$$

$$w^* = \frac{w}{-\frac{\left(\frac{L}{2}\right)^2}{\mu}\frac{dp}{dz}} \tag{4.2}$$

$$y^* = \frac{y}{\left(\frac{L}{2}\right)} \qquad x^* = \frac{x}{\left(\frac{L}{2}\right)} \tag{4.3}$$

$$\alpha_n = \frac{(2n-1)\pi}{2} \tag{4.4}$$

The solution for flow in a square channel is shown in Equation 4.1. The solution is obtained by analyzing ¼ of the channel and solving the differential equations with the separation of variables technique. In Equation 4.2 the non-dimensional velocity in the z direction is denoted by w*. The non-dimensional x and y coordinates are shown in Equation 4.3 as x* and y*. These are normalized by one-half the length (L/2) of a channel side because the origin is placed at the center of the channel cross section. Using this solution, a direct comparison of lattice Boltzmann results and analysis of errors can be accomplished.

Lid-Driven Cavity Flow

Lid-driven cavity flow in a 2D square cavity and a 3D rectangular cavity is performed. This computational case has been used by numerous researchers in the past as a benchmark for CFD codes. The flow within a closed cavity is driven by the lid of the cavity moving from left to right. At a Reynolds numbers around 1000, three characteristic vortices are visible. For this study, a Reynolds number of 1000 is used. As with the Poiseuille flow case, the different boundary treatments are tested with different boundary distances. The boundary distance Δ represents the distance from the node to the cavity boundary on all sides. The resolution and lattice spacing is varied while keeping the Reynolds number constant. The distance to the boundary Δ changes from case to case. Figure 4.3 illustrates the 2D lid-driven cavity flow geometry. In the figure, U_1 is the velocity of the lid moving from left to right and the length of the sides of the cavity is L. The 3D geometry is similar with the extension of the flow cavity and boundaries in the z direction creating a rectangular flow domain. Also in the 3D case, end walls are added to enclose the flow in the cavity.

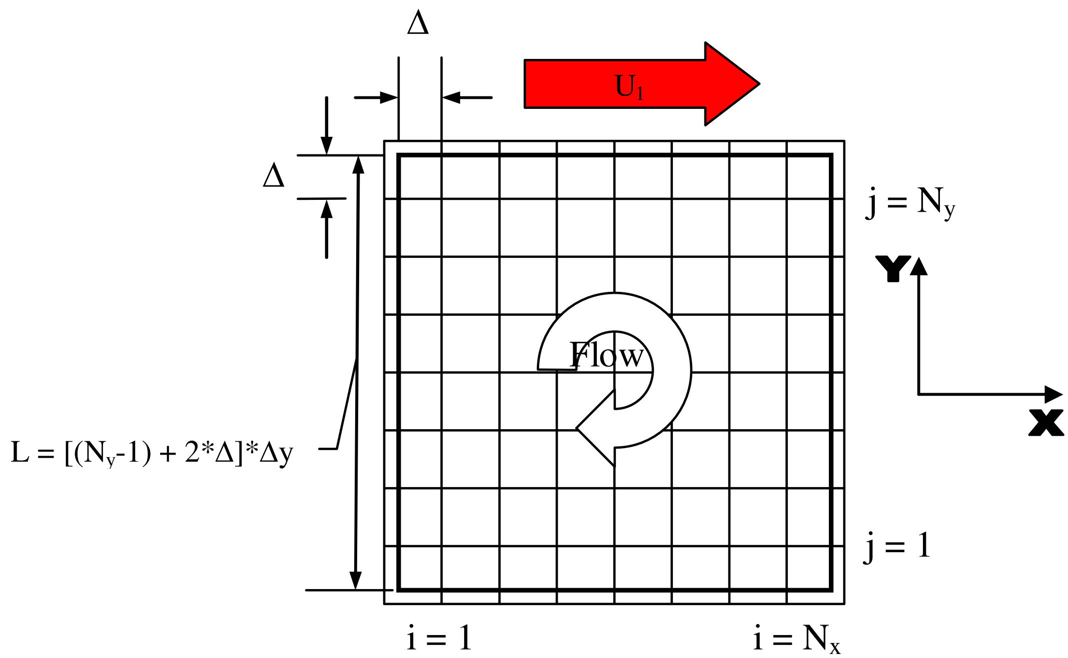


Figure 4.3 – 2D Lid-Driven Cavity Flow Geometry

Previously, Lin & Lai (2000) tested their lattice Boltzmann code using the 2D lid-driven cavity flow case. The results from their computations were compared against previously published data. The results for the vortex positions and velocity profiles along the centerlines compared favorably. In this study, the results from the lattice Boltzmann code are compared directly against 2D and 3D results generated by the commercial CFD package FLUENT. Analysis will include velocity plots across the cavity. The direct comparison of velocity results will also allow the deviation in velocities to be obtained and expressed as an RMS error. The convergence of the solution as the grid resolution increases will also be of interest.

Flow in an Annulus

The flow in an annulus is examined in both 2D and 3D geometries. This flow is driven by the movement of the inner and outer annulus walls. This geometry is similar to the geometry of the RWV bioreactor that is used for the dynamic bioreactor simulations. Figure 4.4 is an illustration of the 2D annulus geometry. In the figure, R_1 and R_2 are the radii of the inner and outer annulus walls. The angular velocities of the walls are ω_1 and ω_2 respectively. In the case of the 3D annulus, the annulus is extended in the z-direction and stationary end walls are added.

For these particular numerical studies, the accuracy of the boundary treatments is tested by varying several parameters. For the annulus case, the MDI, Bouzidi (BZ), Yu (YU), and shifted bounce-back (SBB) boundary treatments are tested. For the shifted bounce-back treatment, the curved boundaries are approximated with a stair-step boundary. Results should give a good indication of how much more accurate the interpolation methods are with respect to the simple shifted bounce-back treatment. While the overall physical dimensions of the annulus will remain the same, resolution and lattice spacing are varied for different grid resolutions. Results for varying resolutions can be analyzed for a reduction in RMS error as the grid resolution increases, or becomes finer.

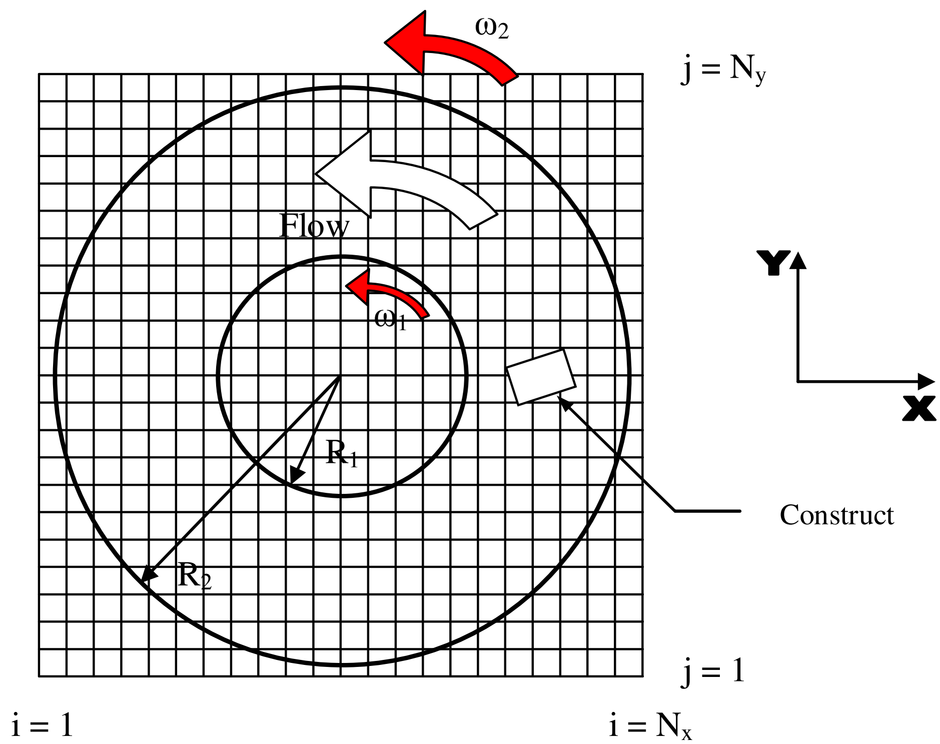


Figure 4.4 – 2D Annulus Case Geometry

Additional parameters are varied for the annulus flow case. Initially, the Taylor-Couette flow case is done. Results from this case are used to compare boundary treatments for accuracy and robustness. From there, the addition of a single stationary construct is also explored. Solutions for 2D flow around circular, square, rectangular, and elliptical constructs are calculated. In 3D, flow around a sphere, cube, disc, and ellipsoid constructs are examined. These results give an even better comparison of the accuracy of boundary treatments because more interesting flow patterns around the construct can emerge. Other parameters that are varied include kinematic viscosity of the fluid, wall velocity, and the density value used in the boundary treatment. Various viscosities are explored to examine stability and accuracy. Previously, Mei et al. (2000)

observed some instability in the MLS boundary treatment for certain values of the relaxation parameter τ and boundary distance Δ. Varying the kinematic viscosity (effectively varying τ in Equation 2.9) and wall velocity should help reveal unstable behavior due to the boundary treatment. The density used at the boundary is varied for the MDI, Bouzidi, and Yu boundary treatments. Previously, the average flow density has been used in momentum exchange (Equations 2.42 and 2.43) by Ladd & Verberg (2001) and Yu et al. (2003). The MDI treatment uses the local node density for momentum exchange. All interpolation boundary treatments are tested using both the average density and the local node density for a specific case. Evaluation of results is structured similarly to the lid-driven cavity flow case with a comparison to a FLUENT solution.

4.1.2 Grid Refinement Studies

The current implementation of grid refinement allows for the coarsening and refining of a grid. The grid refinement studies presented here test both functions, but mainly examine the refining of the grid. Refinement of the grid in more interesting areas of the flow helps reduce computation time and should increase accuracy of the solution. Hence if the method of the refinement of the grid in the lattice Boltzmann is accurate, it would greatly assist in future, more complex computations. The accuracy of the grid refinement is verified by comparing results with either analytical solutions or solutions done with FLUENT as in the boundary treatment studies. The grid refinement studies presented here include Poiseuille flow and lid-driven cavity flow with an irregular grid, refinement in lid-driven cavity flow, and refinement for flow in an annulus.

Irregular Grids for Poiseuille Flow and Lid-Driven Cavity Flow

The primary purpose for these case studies is to test an irregularly spaced or coarsened grid by using the current interpolation methods, namely the MDI treatment, to find necessary PDF values after streaming (Illustrated in Figure 3.12). Poiseuille flow driven by a body force in a channel is again examined in a 2D geometry. Geometry and flow parameters are similar to the ones used in the Poiseuille flow cases for the boundary treatment studies. The boundary distance Δ is fixed to neighboring nodes for all channel walls. However, the grid is made irregular by positioning nodes at varying intervals across the channel rather than at the regular lattice spacing. For the 2D case, nodes not directly bordering the channel walls are randomly placed within a $\pm$ 0.3Δy space of their normal grid position. The velocity profile across the channel is directly compared with the analytical solution. Figure 4.5 illustrates the irregular spacing of nodes across a 2D channel. For the second study, a coarsened grid is used for the 2D case of lid-driven cavity flow at Reynolds number of 1000. Nodes are spaced at 2Δx, 4Δx, 2Δy, and 4Δy away from their neighbors at various locations in the grid. Figure 4.6 illustrates the coarsened spacing of nodes in the 2D cavity. This coarsening is similar to the coarsening done for channel flow by He et al. (1996), except that linear interpolation was used by He et al. and a more complex flow is being attempted here. The solutions using the irregular and coarsened grids are directly compared against the FLUENT solution.

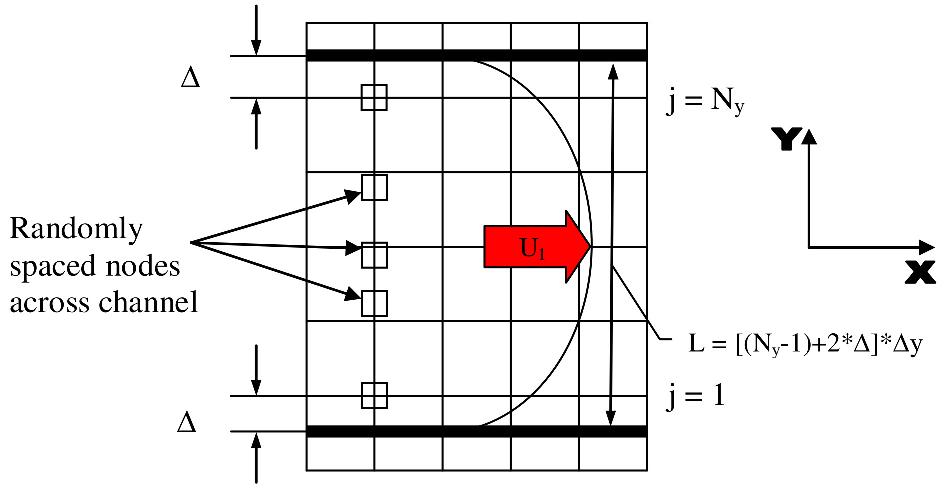


Figure 4.5 – 2D Irregular Grid Channel Flow

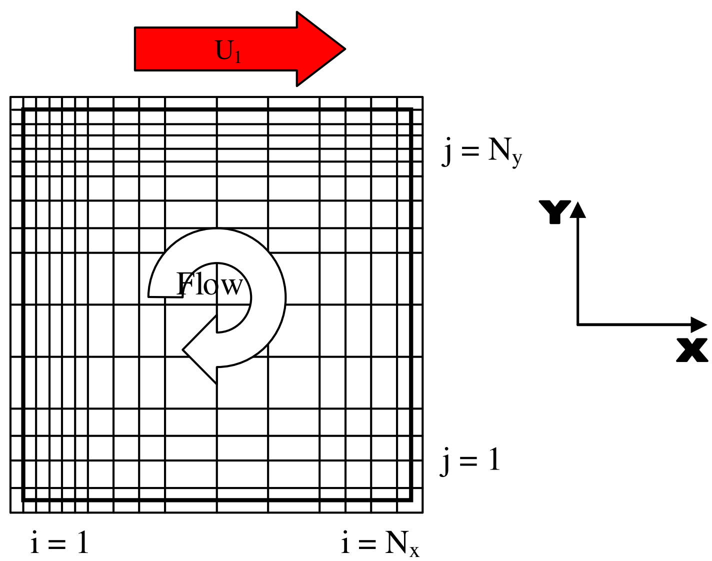


Figure 4.6 – 2D Coarsened Grid for Lid-Driven Cavity Flow

Grid Refinement for Lid-Driven Cavity Flow

The refinement of the grid for 2D lid-driven cavity flow is examined at a Reynolds number equal to 1000. Similar to the study by Lin & Lai (2000), several areas of a cavity are refined. The studies examine areas in the cavity where high velocity errors values typically occur, which includes the upper left and upper right corners. The main grid resolution is varied from relatively coarse to fine, while the grid resolution of the sub-grids is varied as well. Other flow parameters remain constant since testing the grid refinement is of primary interest. The solutions on the sub-grids are compared with the solution produced by FLUENT. Figure 4.7 shows the areas of grid refinement for the lid-driven cavity flow case. This study should assist in testing the accuracy of the present grid refinement implementation (Illustrated in Figure 3.13) as velocity values will be analyzed and compared directly. The varying of resolutions on both the main and sub-grids will also lend insight onto error convergence for both domains.

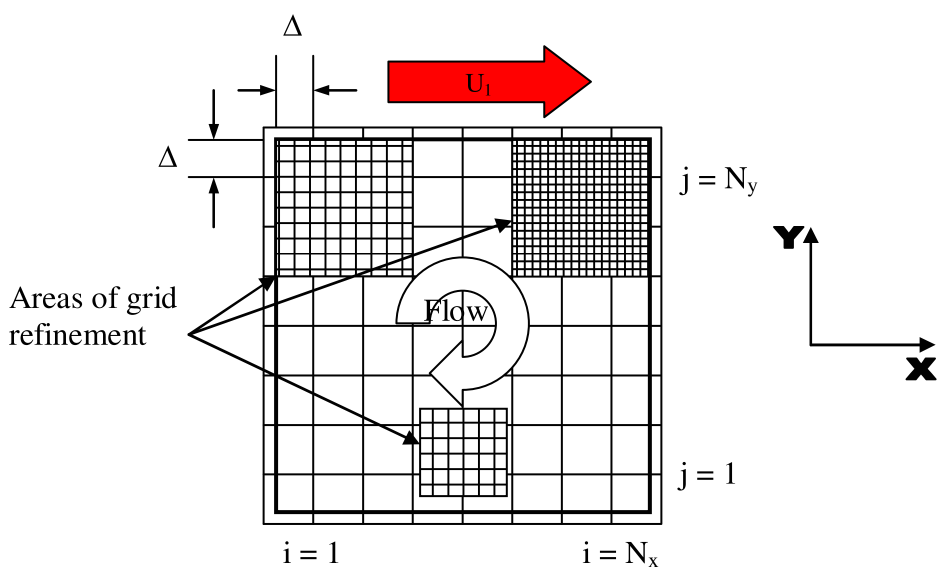


Figure 4.7 – Grid Refinement for the 2D Lid-Driven Cavity Flow Case

Grid Refinement for Flow in an Annulus

The last study presented examines the refinement of the grid for flow in an annulus with a stationary construct. The geometries used are the 2D and 3D annulus geometry previously used in the boundary treatment studies. The area of refinement surrounds the stationary construct because the flow around the construct is of primary interest in this case. Figure 4.8 illustrates the position of the sub-grid with respect to the stationary construct.

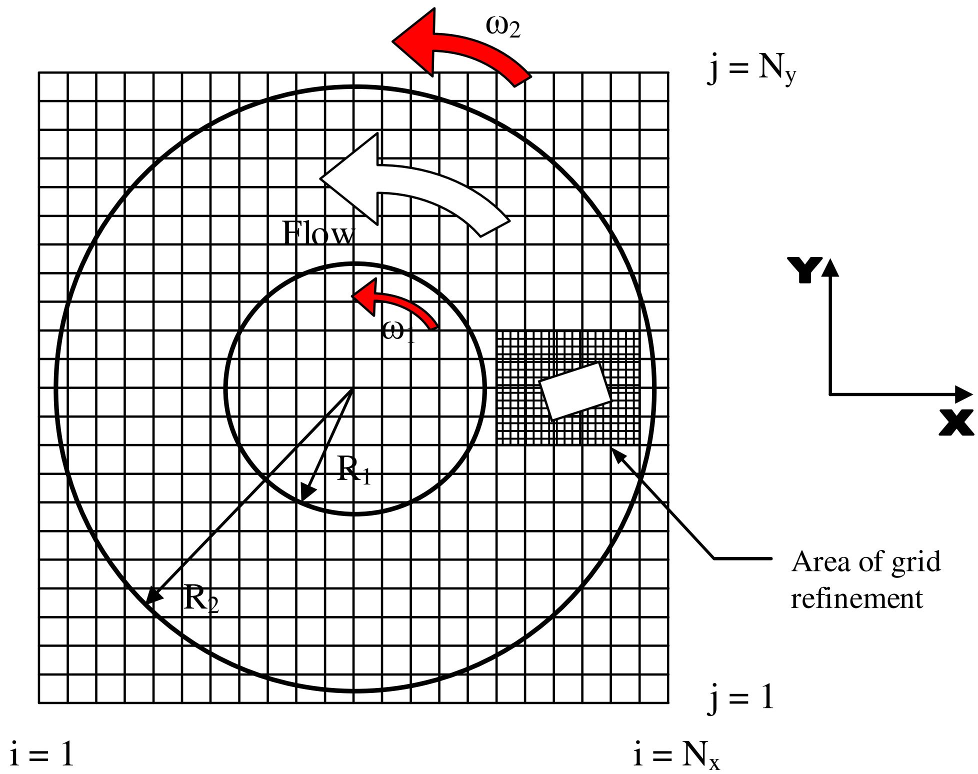


Figure 4.8 – Grid Refinement for the 2D Annulus Case

For the 2D case, the refinement is done for the circular, square, rectangular and elliptical construct cases. Similarly, for the 3D case the refinement is done for the sphere, cube, disc, and ellipsoid constructs. Flow parameters that are varied in this case include the resolution of the main grid, the resolution of the sub-grids, wall velocity and kinematic viscosity. Other parameters remain fixed. However, all boundary treatments are tested across the sub-grids. Accuracy of the refined solutions on the sub-grid is determined by comparing the solution with FLUENT calculations. An accurate grid refinement solution will allow for the calculation of the main flow on a coarse grid while the flow surrounding the construct can be calculated on a refined grid. This is an ideal situation for dynamic simulations because computation time is reduced on the main grid and solutions will also tend to converge faster.

4.1.3 Dynamics Studies

The dynamic studies presented here test the accuracy of the calculation of forces in the lattice Boltzmann, examine shear stress and strain rate calculations, and simulate particle dynamics. The current implementation of the lattice Boltzmann method uses the momentum exchange method to calculate the force on a boundary. Calculated forces on stationary constructs using the momentum exchange method are verified by calculations done in FLUENT. The validation of force calculations is important in that particle motion depends on the forces calculated. The other dynamics studies examine shear stress and shear rates present in fluids. These calculations are also compared with FLUENT results to evaluate accuracy. Finally, two dynamic cases that analyze the motion a particle in flow are performed. The sedimentation of a particle in a channel and the motion of a neutrally-buoyant particle in an annulus are examined. The movement of the particle is compared to either analytic or experimental results.

Force Evaluation on a Stationary Construct

The force on a stationary construct in the 2D and 3D annulus geometry is examined. The geometry is the same as previous annulus studies with a construct (See Figure 4.4). The primary parameters that are examined include force evaluation technique, grid resolution, flow parameters like kinematic viscosity, and construct type. The force evaluation techniques that are compared include the Aidun force evaluation technique (Equation 2.50) along with the current force evaluation technique (Equation 3.1). The calculated force for a given grid resolution and flow parameters is also examined. A change in grid resolution effectively changes the construct radius in terms of lattice spacing while a change in kinematic viscosity should affect the velocities calculated at the boundary and thus accuracy. These parameters are of interest because of conclusions presented in previously published results. Ladd & Verberg (2000) evaluated drag force in a 2D array of cylinders and showed second-order error convergence for increased construct radius. This study should show similar behavior for increased grid resolution. Nguyen and Ladd (2003) published results that exhibited large errors for calculated torques on particles at low viscosity values. This study would hope to test force evaluation at different viscosities to compare results. Lastly, construct type is varied as before with different construct shapes being used. The force results calculated here are compared to FLUENT force and torque values to evaluate accuracy.

Shear Stress Evaluation for Flow in an Annulus

The next set of computations within the dynamic studies evaluates shear stress and shear rates (Equations 3.26 and 3.32) for 2D and 3D flow in an annulus with a stationary construct. Geometry matches previous cases done with the annulus geometry. Similar to the previous case of force evaluation the grid resolution, boundary treatment, and construct type are all varied. The evaluation techniques used for strain rates and ultimately shear stress are based on the lattice Boltzmann (See Appendix B) and the finite difference methods previously described (Equations 3.25 through 3.29). The results from these methods are compared to the results generated by FLUENT. Accurate results for the calculated strain rates and shear stress help in the validation of the code for use in the RWV bioreactor simulations.

Lubrication Force between Two Approaching Particles

The first dynamic case will measure the forces calculated by the lattice Boltzmann code between two particles approaching each other in a channel. The particles are approaching at a fixed velocity in a periodic channel. Previously, Ding & Aidun (2003) performed similar computations using their methods for particles in near contact. Computations were done for both 2D cylinders and 3D spheres. In this study, the flow parameters in the Ding & Aidun studies are matched and the non-dimensional force is examined. In the 2D case, the resolution of the varied while in the 3D case the approach velocity is varied. Figure 4.9 illustrates the geometry of the 2D case. The approaching particles have the same diameter d and approach each other at the same velocity U/2. The 3D case geometry is similar except that the channel is a square channel and the particles are spheres. Additional computations using grid-refinement are performed for the 2D case as well in order to study the difference in using forces calculated from the main grid versus forces calculated using the sub-grid.

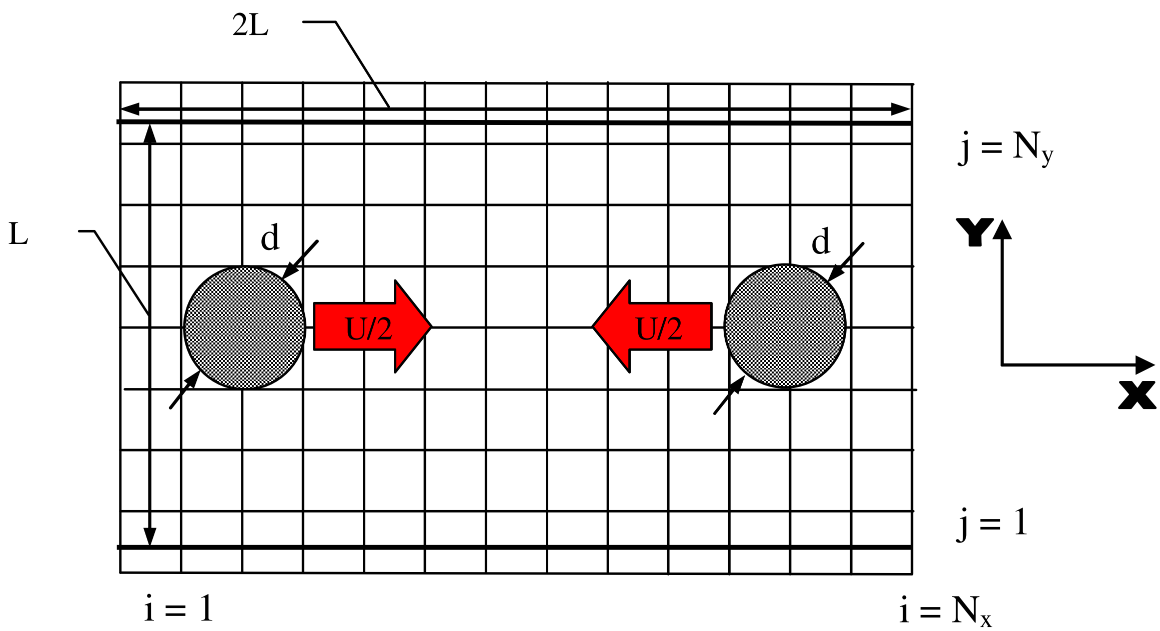


Figure 4.9 – Approaching Circular Cylinders in a 2D Channel

Particle Sedimentation in a Channel

For the second dynamic case, the sedimentation of a circular cylinder or a spherical particle in a long vertical channel is simulated. The particle is released along the centerline of the channel as a force is applied. The Reynolds number is determined from the terminal velocity measured and the particle radius. For this study, Reynolds numbers near zero or Stokes flow will be examined. Previously, experiments have been performed by White (1946) in 2D and by Miyamura et al. (1981) in 3D. In addition, an analytical solution for the 2D case was done by Faxen (1946) and Takaisi (1955) and computational results for the 3D case were presented by Aidun et al. (1998). This study attempts to simulate the same problems in 2D and 3D. Density of the particle in relation to the fluid is initialized to a constant. The effect of the walls on the terminal velocity will be examined and compared to the results of previous studies. In addition, variations on boundary conditions placed on the channel inlet and exit and the effects of using forces calculated from the main grid versus a sub-grid will also be examined. Figure 4.10 illustrates the 2D geometry for the problem. The diameter of the particle is d and the channel width is L. The relation of d to L is varied to examine a range of cases.

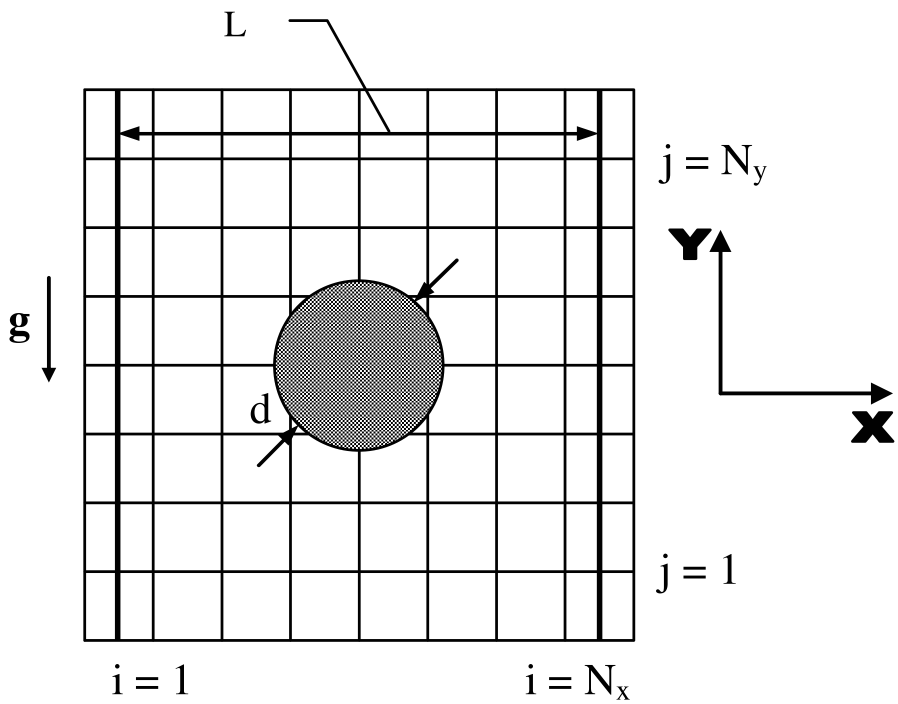


Figure 4.10 – 2D Sedimentation of a Particle in a Vertical Channel

Two Particle Sedimentation in a Channel

The third dynamic case is similar to the previous dynamic case, but adds a second particle to the sedimentation observed in a long vertical channel. In this case, the particles are released offset to the center of the channel and the interaction of the particles while sedimentation occurs is observed. Only the 2D case of circular cylinder sedimentation will be examined. This computation with similar geometry has been previously performed by Feng et al. (1994) and Aidun & Ding (2003). Figure 4.11 shows the geometry of the case where the width of the channel is four times the diameter of the particles. The initial release position of the two particles is one diameter from the left side of the channel ($x_1 = x_2 = 1d$) and the initial spacing between the particles is two diameters ($y_1 - y_2 = 2d$).

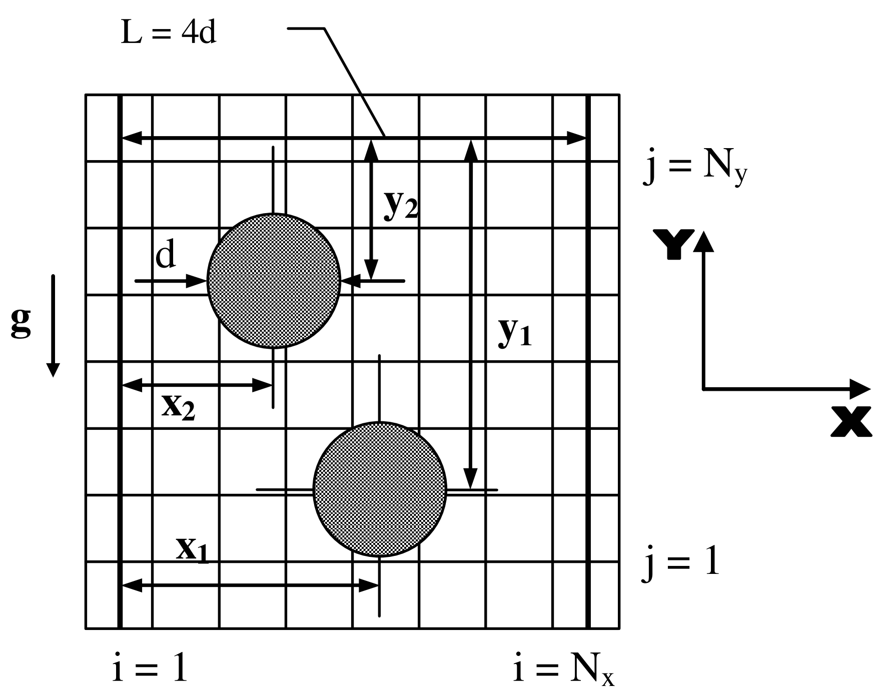


Figure 4.11 – 2D Sedimentation of Two Particles in a Vertical Channel

The particles' paths as they sediment in the channel and the dynamic interaction of the particles are of primary interest. Since the particles come in near contact, this simulation will test the accuracy of the code for an unsteady dynamic situation, particularly the accuracy of the near contact and lubrication forces. An attempt to reproduce the period double behavior seen in the Aidun & Ding study will be made.

However, boundary conditions and lubrication force calculations for this study are not exactly matched to the previous work.

Motion of a Neutrally-Buoyant Particle in an Annulus

For the fourth dynamic case, the motion of a neutrally-buoyant particle is examined within an annulus. For this case, the geometry and flow parameters are similar to previous studies done in the boundary treatment and grid refinement studies for constructs placed in an annulus with the exception of the wall velocities. In this study, the angular velocity of the inner and outer annulus will be set such that case of rigid body rotation is produced. The steady flow solution is first calculated with a stationary construct and then the construct is released. No external or gravitational forces will be applied to the construct which effectively sets the density ratio of the solid to the fluid to one. The 2D and 3D motion of a cylinder or particle using force calculation from a main grid and moving sub-grid will be examined. Previously, Botchewey et al. (2000) visualized the motion of lighter-than-water microcarrier scaffolds in a rotating bioreactor with no inner wall. The results showed that the microcarriers followed a periodic circular motion which agreed well with their theoretical predictions. Although, the geometry and flow parameters of this study are not identical to this previous work, qualitative comparisons can be made. The motion of a circular cylinder particle will be examined in 2D and the motion of a spherical particle will be examined in 3D.

4.1.4 Simulations

The simulations primarily deal with the flow in the RWV bioreactor with a moving construct. These simulations help evaluate if the lattice Boltzmann software is accurate for the range of operational parameters of the RWV bioreactor. These studies are primarily used for the validation of the model for making credible design decisions for the RWV bioreactor. The two simulations that are performed use the 2D and 3D RWV bioreactor geometries. The geometries are similar to the 2D and 3D annulus geometries used previously, but have parameters adjusted to simulate the actual RWV bioreactor environment.

2D Flow in the RWV Bioreactor with a Moving Construct

The geometry and implementation for this case is similar to the geometry and implementation used for the 2D annulus case. There are some differences however for the simulation cases. For the simulations, the dimensions of the bioreactor and fluid properties are matched or scaled to operational conditions for the RWV bioreactor. Also, these simulations have moving constructs rather than the stationary constructs used previously. The construct movement is calculated, as described previously in Section 3.1.3, using a higher-order Runge-Kutta solver for the equations of motion. The inclusion of lubrication forces is necessary for the simulation cases as the construct approach the sides of the annulus. Another consideration for the construct that needs attention is the density of the construct in relation to the fluid. Previously, with a stationary construct this was not considered. In this study, the density of the construct will be matched to the experimental density used by Brown (1998). However, the viscosity of the fluid may not be matched due to the instability of the code. Thus the net body force on the construct may be varied (by varying the gravitational constant g) in order to make the simulations more comparable to the experiments.

Evaluation of the 2D simulations is based on two issues. First, the movements of the construct must be consistent with patterns already described in literature. There are two different and distinct construct movements that have been observed in the RWV bioreactor, the orbiting regime and free-fall regime (See Figure 2.2). The simulation should reproduce these movements. The second issue is if the shear stress experienced by a construct in previous experiments or published in previous studies is compared to the calculated shear stress in the simulations. Although the real RWV bioreactor involves 3D flow, favorable results for these 2D comparisons will help validate the 2D lattice Boltzmann model.

3D Flow in the RWV Bioreactor with a Moving Construct

The 3D simulations of the RWV bioreactor are conducted in the same manner as the 2D simulations. The only major difference is that the 2D geometry is extended in the z-directions with stationary end walls added to the ends of the annulus. Implementation of most of the aspects of the lattice Boltzmann method is essentially the same for 3D. The lattice Boltzmann model used in 3D is different, but most everything else including the boundary treatment and particle dynamics are the same. The evaluation of the 3D simulations is similar to the evaluation in 2D. Main issues addressed are the dynamic construct behavior and the comparison of shear stress simulation results with previous results. However, a third comparison can be made involving how the flow characteristics of the lattice Boltzmann simulations compare with experimental data obtained previously with a RWV bioreactor. 2D PIV visualizations done by Brown (1998) and Neitzel et al. (1998) are used for this comparison. The comparison primarily considers velocity field and general flow characteristics such as shape of the construct wake. Comparable simulation results will validate the 3D lattice Boltzmann model.

4.2 Evaluation

The results of the numerical studies will be evaluated and compared with results generated from analytical solutions, solutions generated by a commercial CFD software package, and experimental results. Visualization of the results is handled by a commercial software packages Fieldview, Excel, and MATLAB. Comparison of analytical solutions and lattice Boltzmann solutions is mostly handled through this software as well. The commercial CFD package that is used in most cases for direct comparison of solutions is FLUENT. The FLUENT solutions calculated are done on very fine grids and will be taken as credible, accurate solutions. Most of the error analysis in comparing lattice Boltzmann solutions with FLUENT solutions is done in MATLAB. Some validation analysis is also done in comparing lattice Boltzmann results with experimental results. These results may either come from previously published research or research done in labs associated with the Georgia Institute of Technology.

4.2.1 Software Methodology

The software that is used in the evaluation of the lattice Boltzmann results includes Fieldview, FLUENT, MATLAB, and Excel. The specifics of how these software packages are used in the evaluation of lattice Boltzmann methods is detailed here.

Fieldview and Excel

Fieldview is a CFD post processor package that was developed by Intelligent Light (Lyndhurst, NJ – www.ilight.com). Fieldview has a variety of capabilities that assist in the analysis of the results. The lattice Boltzmann software exports data in the Plot3D format that is read directly into Fieldview. Fieldview handles all the post processing visualization for all the results including some of the results generated by FLUENT. The visualizations generated from these results can range from simple velocity vector plots or shear stress contour plots to complex streamline calculations and transient animations. As an example, Figure 4.12 is a screen shot of the Fieldview user interface and a velocity vector plot for the 2D annulus case. Data that has been imported and visualized into Fieldview can be exported again for analysis in MATLAB or Microsoft Excel. In general, comparisons of relatively small velocity data sets with analytical solutions are handled using Fieldview and Excel. In addition, small data sets of shear stress and shear rates are handled in the same manner. More complex comparisons from results calculated in FLUENT are done using MATLAB.

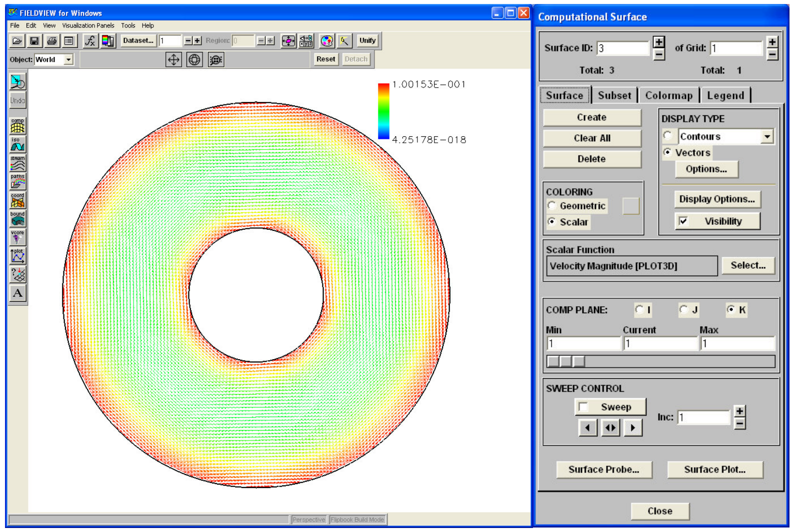


Figure 4.12 – Fieldview Screenshot

FLUENT and MATLAB

FLUENT is a commercial CFD software package distributed by Fluent Inc. (Lebanon, NH – www.fluent.com). FLUENT was first developed in 1983 and has expanded its capabilities as a design and analysis tool with its current release version 6.1. The software is used for simulation, visualization, and analysis of fluid flow, heat and mass transfer, and chemical reactions. It has applications in a diverse number of industries including aerospace, automotive, chemical and materials processing, electronics, and biomedical. In practice, FLEUNT is used in combination with preprocessing software that will build the model and generate a mesh. In this study, GAMBIT is used in conjunction with FLUENT as the preprocessing software. Once the model is built, flow parameters are specified in FLUENT and a case may be run. Some visualization and analysis can be done within FLUENT, but most of that will be handled in Fieldview to have a consistent comparison with lattice Boltzmann results. Figure 4.13 is a screenshot of the FLUENT interface with a visualization of the 2D annulus case.

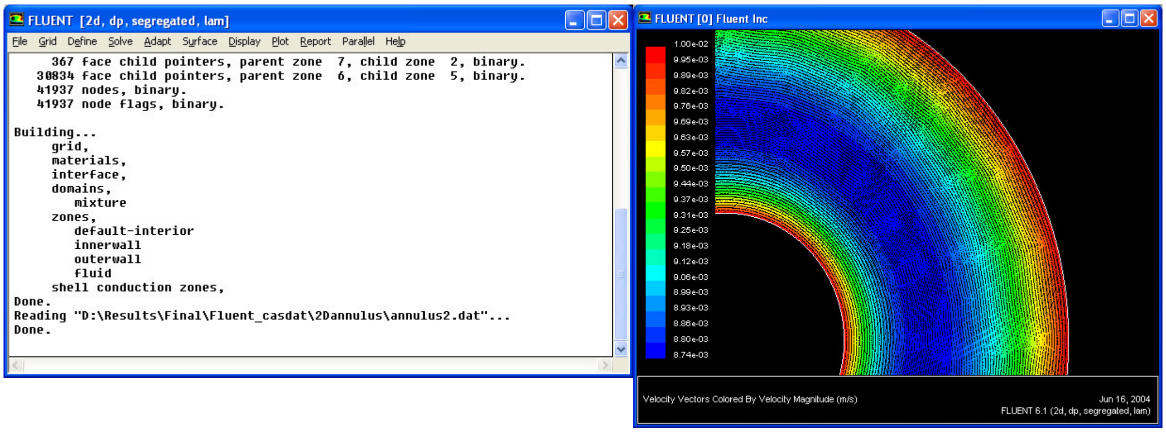
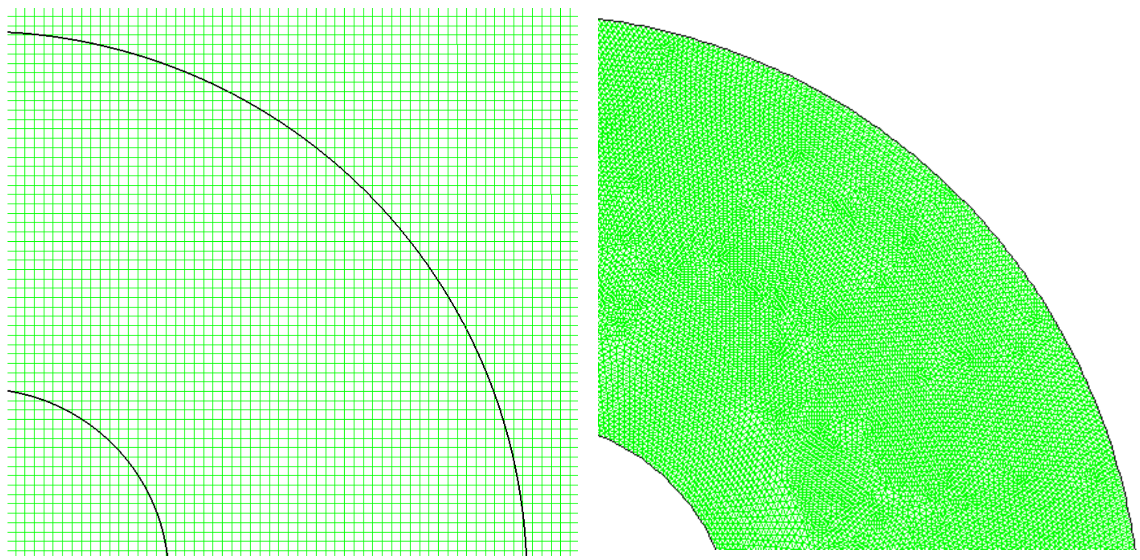


Figure 4.13 – FLUENT Screenshot

Most of the numerical results generated by the lattice Boltzmann software are directly compared to FLUENT results. All of the physical and flow parameters set in the lattice Boltzmann will be identically set in the FLUENT models. No similarity is used, only direct comparison. FLUENT will handle calculations for the lid-driven cavity flow and annulus geometries using very fine meshes. The number of nodes in these meshes will typically be greater than 50K, in order to produce accurate solutions. Figure 4.14 is a comparison of a FLUENT mesh with a typical lattice Boltzmann grid for the 2D annulus case. The FLUENT mesh has over 50K nodes while the lattice Boltzmann resolution is 121x121 nodes. For post-processing visualization, the work is done in Fieldview. However, the velocity error analysis and visualizations of comparisons of FLUENT solutions to lattice Boltzmann results takes place in MATLAB. Data from the

FLUENT solution is used to interpolate a solution on the lattice Boltzmann grid. Once this is accomplished a direct comparison of the values of interest can be done.



Lattice Boltzmann Grid FLUENT Mesh

Figure 4.14 – 2D Grid Comparison

The direct comparison of results at each lattice Boltzmann grid point should result in excellent evidence of the accuracy of the lattice Boltzmann implementation. The primary values that are directly compared are the components of velocity. Other comparisons for FLUENT shear stress and force are made but are not processed by MATLAB. The calculated force on a construct can be calculated in FLUENT and will be compared to lattice Boltzmann results in the dynamic studies using Excel. In order to ensure the convergence of the FLUENT results used in comparisons, computations are run until residual velocity and continuity values converge to a steady value around 10^{-11} or less. In addition, the meshes are then refined by at least a factor of 2 and the computations are run again until converged to make certain that the FLUENT solutions are not grid dependent. For the simulation studies, results with a moving construct are compared to FLUENT moving mesh calculations as well. FLUENT will primarily help verify the accuracy of the implementation for most of the numerical studies. It also will serve to help validate the simulation results; however the comparison of results with previous experimental results is also considered in validating simulations results.

4.2.2 Analysis Methodology

The results are presented and evaluated in three different fashions. Each type of evaluation has its own purposes in testing the accuracy of the model. Visualization, error analysis, and comparison to previous works are all used in the analysis to get a complete picture of the performance of the code. Ultimately, positive results from all aspects of the analysis should verify the accuracy and validate this implementation of the lattice Boltzmann method.

Visualization

The visualization of the results is done in several different ways. The use of line plots, surface plots, contour plots, and streamline visualizations aid in qualitatively evaluating whether or not the results from the lattice Boltzmann match those of the analytic or FLUENT solution. These visualizations techniques have advantages and disadvantages. The primary advantage is that visualizations can provide a quick snapshot of whether or not the results are in the ballpark as the analytic or FLUENT solution. For example, this is the case when absolute velocity error is plotted as a surface in the grid refinement studies. Comparing the maximum absolute error between the main grid and sub-grid can give an initial indication of how well the sub-grid reduces error. This is also the case with streamline visualizations of the flow around a stationary construct in the annulus. Comparing the wake characteristics of the results with the wake characteristics of the FLUENT solution can indicate whether or not the code is giving reasonable results. The disadvantage of using these visualizations is that they often cannot quantify how well the code is working. For example, comparing contour plots of shear stress or absolute velocity error surface plots between different boundary treatments contributes little to conclusions on which boundary treatment is working better on the whole. Thus, the visualizations are great for an overview of the data, but lack in quantifying results when comparing methods.

For the results presented, a variety of visualizations are done for specific cases. Absolute velocity error surface plots are used as an initial evaluation of methods and also are used to draw general conclusions about data. These are done for most of the boundary treatment and grid refinement studies in order to detect areas of high error within the geometry. Velocity line plots are only presented for the lid-driven cavity flow because they often cannot easily show differences between boundary treatment methods. These plots often end up looking similar on large scales. Streamline visualizations are used to illustrate general characteristics of flow containing wake regions and vortices. Contour plots of shear stress and shear rate are done for qualitative evaluation of areas of high shear rates in the studies involving constructs. Shear stress and shear rates line plots are also used for these studies near the surface of the construct because they help quantify these important parameters in crucial area of geometry. In general, the visualizations presented in the results will not be comprehensive in scope, but will represent the general traits and characteristics of the results.

Error Analysis

The error analysis is primarily applied to the velocity results. The analysis involves looking at three key issues: the behavior of the RMS error as grid size is changed, the RMS error magnitudes, and the percent relative error (or percent error) of individual nodes within the flow. The RMS x and y velocity error is calculated by using equations similar to 2.63 where the lattice Boltzmann results are compared directly with FLUENT results at each grid point. The RMS error value thus gives a type of average error for the nodes included in the calculation. The percent error is calculated using a standard percent relative error formula with the FLUENT velocity solution serving as the true velocity value. This gives a more local result rather than an average value. The advantage of examining the error is that the results are more objective as opposed to the visualization which can be subjective. The disadvantage is that the error analysis may yield good results when the actual flow produced may be unphysical.

The RMS error is examined in nearly all of the numerical studies in order to quantify results. The behavior of the RMS error as grid size is changed helps determine the error characteristics of the methods being tested. The simple Poiseuille and Taylor Couette flow studies help determine this behavior in a simple geometry, but this is also tested for more complex flow situations. For example, the behavior of the error as the grid is refined in the grid refinement studies is important in evaluating the refinement method itself. The magnitude of the RMS error is important in determining relative accuracy between lattice Boltzmann methods like the boundary treatments. The absolute RMS error is also correlated to percent error in order to draw conclusions about accuracy.

The velocity percent error for each node is calculated for most of the numerical studies where RMS error is examined. The percent error is presented as a percentage of nodes that are within certain threshold limits. In this study, the results are presented as the percentage of nodes that have a percent error <1%, <5%, and <10%. Thus, the results reveal a distribution of nodes and how accurate those nodes are with respect to percent error. This data is presented in tables. The last issue with percent error is that some nodes are not included in the calculation because of their very low velocity value. When the FLUENT velocity falls below the order of the error that the lattice Boltzmann can resolve, the percent error can be inflated artificially high. Thus, some nodes are excluded from the calculation. The criterion used for exclusion is that the FLUENT velocity is less than $1\text{x}10^{-4}$ times the maximum characteristic velocity.

When looking at all three of these error measures, the performance of the methods can be quantified and compared much more easily. Primarily, the boundary treatments will be compared with respect to RMS error and percent error. In addition, trends in these values will help evaluate other aspects of the code such as the grid refinement. However, error analysis alone cannot fully evaluate the performance of the implementation.

Comparison with Previous Results

When possible, the present lattice Boltzmann results are compared to previously published work. The results from the dynamics studies involving particle sedimentation are compared to the results of White (1946), Miyamura et al. (1981), and Faxen (1946). The results generated by the simulation studies are compared with previously published experimental results. Previous work by Brown (1998) and Neitzel et al. (1998) are used as a validation to the lattice Boltzmann simulation results. As stated earlier, the parameters for the simulation studies attempt to closely match those used in the operation of the RWV bioreactor. Those simulation parameters also mirror the parameters used in the experimental studies. In general, consistency in flow patterns and flow characteristics are evaluated. The lattice Boltzmann simulation should reproduce the construct behavior and flow patterns around the construct seen in previous results. Evaluation is more subjective than objective because of the complex nature of the problem. This contrasts with the error analysis, but evaluates results with more of an emphasis on the physics of the model.

4.3 2D Results

The 2D results were obtained using the D2Q9 Lattice Boltzmann model with the methods presented in Chapter 3. In order to characterize how well the model performs, numerical studies were done. The primary numerical studies presented here tested methods involving the boundary treatments, grid refinement, and application of forces and dynamics. Once these components of the Lattice Boltzmann code were verified with the studies, the accuracy of the code as a whole could be better qualified. Following the numerical studies, results from the full simulations of the RWV bioreactor geometry are presented. These results aim to validate the use of the code for this particular bioreactor problem. Appendix C contains the results from the numerical studies and simulations done for the 2D results and is referenced throughout the presentation of the 2D results.

4.3.1 Boundary Treatment Studies

There were several numerical studies done to examine the accuracy of previous boundary treatments presented in literature and the newly proposed MDI boundary treatment. The accuracy of the shifted bounce-back (noted as SBB), the Bouzidi quadratic interpolation (noted as BZ), and the Yu linear interpolation (noted as YU) boundary treatments were examined and compared to the MDI boundary treatment. These studies used some simple benchmark cases with different geometries to evaluate the boundary treatments.

Poiseuille Flow with shifted boundaries

The body force driven flow in a channel was examined with shifting boundaries. The body force driven flow for all cases had the maximum centerline velocity set at U_1 = 0.1m/s. For simplicity, average density was set at 1.0kg/m^3 and the relaxation constant was set at $\tau = 1$ for all cases. Computations for five cases with different boundary positions were performed. For the initial case the boundary distance Δ (See Figure 4.3) was set at 0.5. Only the shifted bounce-back (SBB) boundary treatment was used for this case. The boundary distance was then set to $\Delta = 0.6$, $\Delta = 0.7$, $\Delta = 0.8$, and $\Delta = 0.9$ for the next four cases D1, D2, D3, and D4 respectively. For these situations, the MDI, Bouzidi quadratic, and Yu linear (MDI, BZ, and YU) boundary treatments were examined. Therefore a total of thirteen computations were run for the five boundary distances. The total width of the channel L was varied by changing the resolution of the computation and keeping the lattice spacing Δy constant at 0.1m. Table 4.1 shows the horizontal channel flow (HCF) grids used along with their respective resolutions. A total of 65 computations (5 resolutions with 13 computations each) were performed.

Table 4.1 – Lattice Boltzmann Parameters for the 2D HCF Grids

Grid	Resolution	Lattice Spacing Δy (m)	Channel Width L (m)	Boundary Distances Δ	Boundary Treatments Tested
HCF1	21x3	0.1	0.3	0.5	SBB
HCF2	21x6		0.6	D1 = 0.6	MDI,BZ,YU
HCF3	21x9		0.9	D2 = 0.7	MDI,BZ,YU
HCF4	21x12		1.2	D3 = 0.8	MDI,BZ,YU
HCF5	21x15		1.5	D4 = 0.9	MDI,BZ,YU

The RMS error was calculated for the y-velocity profile by comparing the computed results with the analytical solution (See Equation 2.63). The results for RMS error versus $1/L^2$ are presented in Figures C.1 to C.4 for the four cases D1, D2, D3, and D4. In each of these figures, the RMS error for the shifted bounce-back case is plotted as a baseline reference for accuracy. Linear trend lines have been added to help visualize the data. Also, note that all of the graphs are scaled identically on the RMS error axis for easy comparison. To further illustrate the error characteristics of the boundary treatments as the channel shifts, Figure C.5 plots the y-velocity profile for the HCF1 grid where only three nodes span the channel width. The analytic solution is plotted along with the shifted bounce-back results and the results from the four cases D1, D2, D3, and D4 for all boundary treatments. Both of these error visualizations are useful in differentiating the basic behavior of the boundary treatments. Lastly, Table C.1 summarizes the RMS error results for the various cases. The data presented in Table C.1 essentially restates conclusions that can be made from the figures. First, all of the boundary treatments display second order error convergence. Second, the SBB treatment is most accurate for halfway boundaries, while the second order interpolation treatments (MDI and BZ) fare better than the linear (YU) treatments. In addition, the RMS error is the lowest when the boundary distance is at its midpoint ($\Delta = 0.5$) and increases for the interpolation treatments as it moves away from this point. Thus, some basic behavior of the interpolation treatments is illustrated by this shifted channel case.

Lid-Driven Cavity Flow

For the lid-driven cavity flow (LDCF) case, a set of numerical studies were performed that varied resolution and lattice spacing. The Reynolds number for these cases was set constant at 1000 for a square cavity with a side length L = 24.18dm = 2.418m. For Re = 1000, the kinematic viscosity was held constant at $\upsilon = 0.01 dm^2/s = 0.0001 m^2/s$, the lid velocity was set at U1 = 0.413565dm/s = 0.0413565m/s, and the density was set to 1kg/dm3 = $1000 kg/m^3$. See Figure 4.3 for an illustration of the geometry. The four grid resolutions tested were 51x51, 101x101, 151x151, and 201x201 and were labeled LDCF1, LDCF2, LDCF3, and LDCF4 respectively. Table 4.2 summarizes other key parameters for the numerical studies done at these four resolutions.

Table 4.2 – Lattice Boltzmann Parameters for the 2D LDCF Grids

Grid	Resolution	Lattice Spacing Δx (dm)	Boundary Distance Δ	Relaxation Parameter τ
LDCF1	51x51	0.48	0.1875	0.5625
LDCF2	101x101	0.24	0.375	0.625
LDCF3	151x151	0.16	0.5625	0.6875
LDCF4	201x201	0.12	0.75	0.75

As resolution changes, all three parameters in Table 4.2, lattice spacing, normalized boundary distance, and relaxation parameter, must be altered in order to keep the same physical geometry of the problem. This aided in the comparison of the lattice Boltzmann results with a single FLUENT simulation. For each resolution, the SBB, MDI, BZ, and YU boundary treatments were tested. Therefore 16 computations (4 resolutions with 4 boundary treatments each) were performed for this numerical study. For the shifted bounce-back boundary treatment, boundaries were treated as if they were a half lattice spacing away or $\Delta = 0.5$. All other boundary treatments could handle the varying boundary distance. All of these results were compared to a lid-driven cavity flow FLUENT simulation of the same geometry done with a mesh containing over 100K nodes.

For the LDCF results, several types of visualizations were performed. Figures C.6, C.7, and C.8 are streamline visualizations for the lid-driven cavity flow case where Re = 1000. In Figure C.6, the streamlines as calculated by Fieldview for the FLUENT solution are shown. For Figure C.7 and C.8, the streamlines for the low resolution case LDCF1 (51x51) and high resolution case LDCF4 (201x201) are shown. The red arrows indicate the direction of the lid movement. These streamline visualizations indicate that the lattice Boltzmann code is improving with an increase in resolution and compares well qualitatively to FLUENT. Another type of visualization performed is velocity line plots. Figures C.9 to C.12 are plots of the normalized x and y velocities at the centerlines of the cavity. The velocities are normalized by the lid velocity U_1, while the position is normalized by the cavity dimension L. The blue axes and lines are for the x-velocity, while the black axes and lines are for the y-velocity. These plots show the results of the

LDCF1 and LDCF4 grid cases versus the FLUENT solution for the four boundary treatments tested. The line plots give an indication of the velocity accuracy of the code. As stated previously, it is difficult to quantify how well one boundary treatment outperforms another. The last visualization performed for the LDCF cases is absolute error plots. These plots are a better indication of the areas within the cavity where the most error occurs rather than just providing a snapshot like the line plots. Figures C.13 and C.14 are plots of the absolute error for the x and y velocities, respectively, for the LDCF1-SBB case. Note that the lower left corner of the cavity is located at the absolute coordinates (0dm, 0dm). From the plots, it is clear that the areas of highest error are in the upper left and upper right corners of the cavity. This is expected since this is where the singularities occur. To evaluate the effect of increased resolution on absolute error, Figures C.15 and C.16 shows the errors for the x and y velocities, respectively, for the LDCF4-SBB case. The effect of boundary treatments can be seen by comparing these figures with Figures C.17 to C.19. These are plots of the x-velocity absolute error for the LDCF4 case using the MDI, BZ, and YU boundary treatments respectively. From these plots, it is clear that there is a decrease in absolute error when increasing resolution and when comparing the SBB treatment to the MDI, BZ, or YU interpolation treatments.

To further quantify how well the lattice Boltzmann results compare to the FLUENT results, the x and y velocities are analyzed. The FLUENT solution velocities are interpolated onto the lattice Boltzmann grid using MATLAB and are compared directly with the lattice Boltzmann results. The RMS error for the x and y velocities are calculated for all 16 computations, which included four resolution cases with four boundary treatments. The RMS error results for the x-velocity and y-velocity are presented graphically in Figures C.20 and C.21. The RMS error for both x-velocity and y-velocity decrease with increasing resolution in these figures which is expected. What is not expected is that there is no consistent result concerning boundary treatment. In other words, a different boundary treatment has the lowest RMS error for each resolution. Analysis of the trends in percent error can also help characterize the accuracy of the code. In Tables C.2 and C.3, data for the percent error for the x and y velocities is presented. The percentage of grid points or nodes with a percent error of <1%, <5%, and <10% is listed for each resolution case and each boundary treatment. These percent error results supplement the RMS error results and can differentiate the performance of the boundary treatments. In general, it can be seen that there is an increase in the percentage of nodes with low percent error when there is a decrease in RMS error. However, the difficulty in analyzing this data is comes in part from the fact that there are singularities at the upper left and right corners of the cavity that may be causing inconsistencies. Despite these singularities, the data does yield some general insights about the current lattice Boltzmann implementation.

Flow in an Annulus

Flow in an annulus is a geometry that essentially matches the geometry of the RWV bioreactor. There were several numerical studies done with this geometry in order to examine the accuracy of the code with respect to curved boundaries, varying viscosity and maximum velocity, and constructs placed within the flow. These numerical studies were done with an annulus having an inner radius of R_1 = 0.609dm = 0.0609m and an outer radius of R_2 = 1.752dm = 0.1752m (See Figure 4.4 for illustration). The absolute coordinates of the annulus center was located at (1.8dm, 1.8dm). The velocity of the walls was varied as was the viscosity of the fluid for several different cases examined. Table 4.3 shows the values of viscosity and velocity used for each of the nine cases calculated. For all the cases, the density of the fluid was kept constant at $1kg/dm^3$ = $1000kg/m^3$.

Table 4.3 – Velocity and Viscosity values used in 2D Annulus Cases

Case	Wall Velocity $V_1 = V_2$ (dm/s)	Kinematic Viscosity υ (dm^2/s)
CAS1	0.05	0.01
CAS2	0.05	0.001
CAS3	0.05	0.0005
CAS4	0.1	0.01
CAS5	0.1	0.001
CAS6	0.1	0.0005
CAS7	0.2	0.01
CAS8	0.2	0.001
CAS9	0.2	0.0005

The range of viscosity values combined with the constant density used in these studies is intended to represent properties of common fluids. Some types of oils have kinematic viscosities at the high end of the range, around $0.01dm^2/s = 0.0001m^2/s$. At the lower end of the viscosity range, $0.0005dm^2/s = 0.000005m^2/s$, the fluid starts approaching the viscosity of water which is around $0.0001dm^2/s = 0.000001m^2/s$. Thus, these cases attempt to cover a practical range of kinematic viscosities for testing the lattice Boltzmann implementation. The range of velocities tested is intended to test how well the lattice Boltzmann performs as the computational parameters move away from a low Mach number situation. As stated earlier in Chapter 2, in order for the equilibrium particle distribution formulation to hold the Mach number ($Ma = u/c_s$ where c_s is the speed of sound) must be small. Thus, there is typically some instability with higher velocities with computations using the lattice Boltzmann method.

There are three lattice resolutions that are tested with the annulus geometry. The units for the lattice spacing were chosen as dm, in order to accommodate the simulation of the more common fluids previously mentioned. The geometry chosen for these studies is approximately that of the RWV bioreactor excepted scaled up six times. The total length that spans the lattice Boltzmann grids is 3.6dm or 36cm. The typical diameter of the RWV as seen in previously published works is around 6cm. Table 4.4 details some of the geometry parameters associated with the three lattice resolutions.

Table 4.4 – Lattice Boltzmann Parameters for the 2D Annulus Grids

Grid	Resolution	Lattice Spacing Δx (dm)
ANN1	41x41	0.090
ANN2	81x81	0.045
ANN3	121x121	0.030

For the first numerical study, the nine cases are computed for each of the three grid resolutions: 41x41, 81x81, and 121x121. In addition, the four boundary treatments will all be tested which comes to 108 total computations (9 cases with 3 resolutions using 4 boundary treatments). Since the geometry of the problem is kept constant across the three resolutions, the lattice spacing Δx will differ for each grid. Also, the relaxation constant τ for each case and grid will be different since the kinematic viscosity is held constant with changing lattice spacing (See Equation 2.9). Often the relaxation constant must be within a reasonable range (~0.525 to 1.5) in order to preserve stability in the code as over or under relaxation can cause issues. The results of these computations are compared directly to FLUENT results done on a very fine grid for the nine cases shown in Table 4.3.

Table 4.5 summarizes the stability of the lattice Boltzmann results with respect to the nine cases, three resolutions, and four boundary treatments tested. After initial trials for the geometry chosen, CAS9 results from FLUENT showed that it was an unsteady flow situation that did not converge to a solution and therefore will not be discussed. The stability of the computations is summarized in the table below. In the table, an S stands for all boundary treatments stable, U stands for all boundary treatments unstable, while SBB, MDI, BZ, and YU designate that particular boundary treatment as unstable.

Table 4.5 – Stability of Lattice Boltzmann 2D Annulus Results

Grid\Case	CAS1	CAS2	CAS3	CAS4	CAS5	CAS6	CAS7	CAS8	CAS9
ANN1	S	S	MDI	S	S	U	S	U	N/A
ANN2	S	S	BZ	S	S	BZ	S	S	N/A
ANN3	S	S	S	S	S	S	S	S	N/A

Some general observations for Table 4.5 can be made. First, in general for other parameters holding constant the higher the viscosity or the lower the maximum velocity the more stable the lattice Boltzmann implementation. This is evident when noting the instability of all boundary treatments for CAS6 and CAS8 for the ANN1 grid and with total instability of CAS9 which had the lowest viscosity and highest velocity. Higher viscosity and lower velocity are characteristics of lower Reynolds number flows, which the lattice Boltzmann has performed well with in previous studies. Higher Reynolds number flows are more problematic. Better stability in the case of higher resolution, or smaller lattice spacing, is due to the improved resolution along and better approximation of the curved surfaces in this case. Better resolution allows for less error in interpolation and better approximation of the smooth and continuous particle distribution functions. One last observation is that the lower order boundary treatments are more stable that the higher order treatments for this geometry. This can be seen in the instability of the MDI and Bouzidi boundary treatments in CAS3 and CAS6 for the lower resolution grids. While stability may be a strong advantage for the lower order treatments, their accuracy may be a tradeoff and it must be analyzed further to characterize the lattice Boltzmann implementation.

For the simple Taylor Couette problem, the only visualizations done were the absolute error plots. Figure C.22 and C.23 are typical plots of the absolute value of the velocity error. Specifically, these plots are for the x-velocity and y-velocity error for the ANN2-CAS5-MDI case. In the figures, it is apparent that the area with the largest error peaks is where the radius of curvature is small i.e. near the inner radius of the annulus. This implies that as the radius of curvature approaches the lattice spacing, the lattice Boltzmann produces more error due to the regular grid and the irregular boundary.

The x-velocity RMS error results from the numerical studies involving the Taylor Couette annulus problem are presented. These are essentially identical to the y-velocity RMS error results because of symmetry in the problem. Figures C.24 to C.27 show the RMS error for the ANN3 grid (121x121 resolution) using the SBB, MDI, BZ, and YU boundary treatments, respectively. For each of these figures, results for CAS1 to CAS8 are shown. Note that the z-axis scale for the RMS error is identical for easy comparison across figures. These figures illustrate some important trends in error seen across cases parameters and boundary treatments. Portions of the percent error results for the x and y velocities are listed in Tables C.4 to C.6. Table C.4 lists the CAS5 results which are representative of the results for CAS1 through CAS4. Tables C.5 and C.6 present the results for CAS6 and CAS8 respectively, which illustrate how the boundary treatments perform under a change in Reynolds number. For these cases, differences are observed between boundary treatments at the <1% error threshold.

The next annulus cases performed involved four different constructs placed in the annulus at the 3 o'clock position (See Figure 4.4). These constructs served as a stationary obstacle in the flow. The four construct shapes used included a square (SQ), circle (CIR), rectangle (RECT), and ellipse (EL). The square was oriented with its sides aligned with the x and y axis. The rectangle and ellipse however were placed with their long axis at a 15 degree angle with respect to the x-axis. The diameter or length of the long axis of the construct was set to 0.408dm = 0.0408m. For this geometry, this approximates the size of a real life construct in a RWV bioreactor given the size of the annulus. Also note that at the highest resolution, the diameter of the construct is equivalent to approximately 13 lattice spacings, and at the lowest resolution, approximately 5 lattice spacings. For this part of the numerical study, calculations were first done using the three annulus grid resolutions for the four constructs using the CAS5 flow parameters. In addition, the four boundary treatments were used for a total of 48 computations (4 constructs with 3 resolutions using 4 boundary treatments). These computations should illustrate basic differences in boundary treatments and help evaluate accuracy as the grid is refined. A second set of calculations were done with the finest grid resolution, ANN3, using the CAS6 and CAS8 flow parameters. These resulted in 32 additional computations which should illustrate any degradation in accuracy due to increasing Reynolds number by lowering the viscosity or increasing the wall velocity. As previously, the identical geometries and parameters were used in FLUENT to calculate a solution for comparison.

With a stationary construct placed in the annulus and the CAS5 parameters being used, the flow inside the annulus becomes more complex than the Taylor Couette situation previously analyzed. These flows still converge to a steady solution with a wake region forming behind the constructs. Figure C.28 illustrates the wake region formed behind the rectangle construct that is tilted at 15 degrees using streamlines. The left image is from the FLUENT solution while the right image is from the lattice Boltzmann calculations. The red arrows indicate the direction of wall movement. The other constructs form similar complex wake regions and qualitatively the lattice Boltzmann does well reproducing these wake regions.

Another visualization performed for the annulus with a construct is the absolute error plots. The absolute value of the x-velocity and y-velocity error for an annulus with a rectangle construct is plotted in Figures C.29 and C.30. For these particular plots, the ANN3 grid using the MDI boundary treatment is presented. Again, these plots of the absolute error give an idea of where the most error is occurring in the given geometry. In the case of placing a construct in the annulus, it is clear that the areas with the highest velocity error are located around the construct itself. Specifically, high error is seen at the leading edges of the rectangle and also in the areas surrounding the rectangle's wake region. This is typically the case with the other constructs as well. This is not surprising as the lattice Boltzmann has difficulty with approximating complex boundaries on regular grids especially at relatively low resolution. The magnitude of these errors is much greater than the error seen in the Taylor Couette case (See Figures C.22 and C.23), thus higher RMS error values are expected for these flows.

The x and y velocity RMS errors for velocities were calculated for the annulus construct cases. Figures C.31 to C.33 are representative of the typical RMS error trends seen for the stationary construct calculations. For the following results, the RMS error for the y-velocity is presented because this velocity generally has a higher order of magnitude for this case. Also, the trends and magnitudes for the x-velocity RMS errors are similar. Figure C.31 plots the RMS error for the y-velocity versus all the constructs for the ANN3-CAS5 grid and case. In Figure C.32, the RMS error for the y-velocity is plotted versus the grid resolutions for the four boundary treatments and the CIR construct. For Figure C.33, the RMS error for the y-velocity is plotted versus the three different cases (CAS5, CAS6, and CAS8) run for the EL construct for the ANN3 grid only. In addition to RMS error, the corresponding percent error data is also presented. In Table C.7, the y-velocity percent error results are listed for all the constructs for the ANN3-CAS5 grid and case. The y-velocity percent error data for the ANN-CAS5-CIR cases is shown for all three grid resolutions in Table C.8. The y-velocity percent error data for the three cases (CAS5, CAS6, and CAS8) done for the ANN3-EL grid are presented in Table C.9. These results illustrate trends across construct type, grid resolutions, and case parameters. Increasing resolution decreases error, while increasing Reynolds number increases error. Specifically, increasing wall velocity results in higher RMS error than reducing kinematic viscosity when holding Reynolds number constant. There is no significant difference in error seen between construct types for the grids resolutions used. For boundary treatments, the only real consist result seen is that the

SBB boundary treatment underperforms compared to the interpolation treatments. Lastly, there is a general connection correlation between low RMS error and high percentages of nodes with low percent error. This is most prevalent at the percent error <1% threshold.

The last study performed involved testing the use of local density (or density at the node) versus average flow density for the momentum exchange at a boundary. The primary motivation for testing the density used in the momentum exchange rule (Equations 2.42 and 2.43) is that previously published work has used the average flow density in these equations. This may yield inaccurate results along boundaries, especially moving boundaries. The current implementation of the lattice Boltzmann deals with numerous moving boundaries. Since the local node density is used in this implementation of the lattice Boltzmann the accuracy of this change is examined. A visualization of the absolute y-velocity error is presented for both cases in Figures C.34 and C.35 for Taylor-Couette flow using the ANN2-CAS4-BZ grid and case. Table 5.10 is a comparison of the RMS error results for the same case but across all boundary treatments. This grid and case were chosen because the range of the RMS error was small across all boundary treatments. The visualizations show an increase in error for the average density case (Figure C.35). In addition, the error is more systematic versus the more random error seen in the local density case (Figure C.34). The table clearly shows an approximate three-fold to four-fold increase in RMS error when the average flow density is used rather than the local density.

The results from the boundary treatment studies shed some light on the stability of the boundary treatments and their relative performance. As the Reynolds number increases, the velocity RMS error increases and stability decreases. In general, as grid resolution increases, the velocity RMS error decreases for all boundary treatments. However, the SBB boundary treatment did not perform as well as the other interpolation boundary treatments, especially for the annulus cases.

4.3.2 Grid Refinement Studies

The results of the grid refinement studies are presented here. While efforts were made to test irregular or coarsened grids, the primary goal of refining the grid in this implementation of the lattice Boltzmann is to improve the accuracy of a calculation in areas of interest. Two benefits can be enjoyed through grid refinement: the reduction of computation time and the increased accuracy of velocities around a construct, which leads to better approximations of forces and shear stresses on the construct. The results of the numerical studies in this section are intended to assess the accuracy of the grid refinement implementation in terms of velocities, while part of the results in the next section will assess the accuracy of the force and shear stresses when using grid refinement.

Irregular Grids for Poiseuille Flow and Lid-Driven Cavity Flow

Two attempts were made in testing the ability of interpolation techniques, specifically the MDI treatment, in allowing for irregular grids or coarsened grids. For Poiseuille flow in a channel, an irregular grid was generated by randomly placing nodes a slight distance away from their expected y position (See Figure 4.5). The intent was to test if the MDI treatment could compensate for the irregular position of the node by interpolating the PDF values that would arrive at the nodes after the streaming process. If successful, this may lead to irregular grids that could be more adaptable to boundaries. However, the results indicated that for the D2Q9 model, symmetry in node location should be preserved, as the computations were unstable.

The coarsening of the grid was attempted for the lid-driven cavity flow case. The spacing of the grid was varied at different locations within the cavity (See Figure 4.6). As with the previous case, the intent was to test the ability of the MDI treatment to interpolate post-streaming PDF values. The results indicated that the post-streaming PDF values that were interpolated were not consistent in recreating the flow within the cavity at Reynolds number equal to 1000. This was most likely due to the complexity of the PDF surfaces near the boundaries and the lack of resolution and information available at the boundaries for the interpolation to resolve the PDF surfaces with accuracy. While interpolation techniques still may work on coarsened grids, in this particular case more information is needed at nodes in the areas of coarsening, especially at the boundaries, for better accuracy.

The failure of the current lattice Boltzmann implementation to handle these two cases of irregular grids is not a criticism of the interpolation methods used or the concept of sufficiently smooth PDF surfaces. Rather, the failures are most likely related to the core of the lattice Boltzmann model used and the lack of information available for the MDI treatment. The lattice Boltzmann has been used on unstructured grids by Ubertini et al. (2003), but with a finite-volume formulation not the D2Q9 model. Future research is needed in order to address some of the issues that have caused instability or inaccuracy with these irregular grids in the D2Q9 model. However, the coarsening or use of irregular grids is not necessary in implementing the lattice Boltzmann for simulating the RWV bioreactor. As previously stated, the primary goal of the numerical studies in this section is to test the accuracy of the refinement of the grid, which is practical for the current application.

Grid Refinement for Lid-Driven Cavity Flow

Grid refinement was performed using the previous lid-driven cavity flow computations presented in the boundary treatment studies. The geometry and flow parameters of the LDCF cases were preserved while additional grid refinement studies were done on the LDCF1, LDCF2, and LDCF4 grid resolutions. For each grid resolution, multiple grid refinement computations were performed in order to achieve sub-grid resolutions below that of the finest main grid resolution, LDCF4. A grid refinement calculation involves placing a sub-grid over the main grid and iterating the lattice Boltzmann method on the sub-grid during one main grid time step. Thus, sub-grid computations do not take as much time as finer main grid computations. The other advantage is that the sub-grid can be placed anywhere over the main grid at any time step. Table 4.6 lists the grid refinement computations done for each main grid resolution.

Table 4.6 – Grid Refinement Computations for the 2D LDCF Main Grid Resolutions

Grid	Main Grid Resolution	Lattice Spacing Main Grid Δx (dm)	Refinement Factor	Lattice Spacing Sub-Grid Δx (dm)
LDCF1	51x51	0.48	2	0.24
			4	0.12
			8	0.06
LDCF2	101x101	0.24	2	0.12
			4	0.06
LDCF4	201x201	0.12	2	0.06

The table shows that for the coarsest main grid resolution, LDCF1, three computations are done for the refinement factors 2, 4, and 8. The finest resolution grid, LDCF4, only had one computation performed with a refinement factor of 2. Note that the effective resolution of the LDCF1 sub-grid with a refinement of 8 is identical to that of the LDCF2 sub-grid with a refinement of 4 and the LDCF4 sub-grid with a refinement of 2 because of identical sub-grid spacing. Also note that this particular sub-grid is twice as fine as the finest main grid, LDCF4. Thus, it is seen from the table a set of six separate computations across main grid resolutions were performed to examine the effectiveness of refining the grid.

Three areas of refinement were chosen for the lid-driven cavity flow case: the upper left (UL) corner of the cavity, the upper right (UR) corner of the cavity, and the lower middle (LM) of the cavity (See Figure 4.7). The first two areas were chosen to test the accuracy of the velocities produced by the sub-grid when boundaries are present, while the third area tests the consistency of the sub-grid velocities with the main grid velocities when no boundaries are present. For each of these areas of refinement, the four boundary treatments (SBB, MDI, BZ, and YU) were again tested to highlight the differences in accuracy. Thus, 72 total computations (3 areas of grid refinement with 6 sub-grids using 4 boundary treatments) were completed for this section.

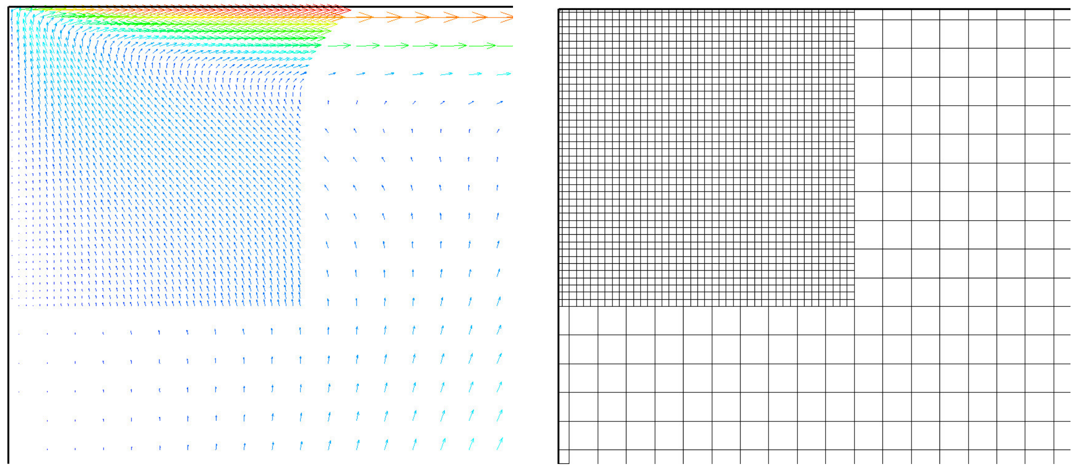


Figure 4.15 – Velocity Vectors and Mesh for Grid Refinement in the LDCF UL Case

Figure 4.15 illustrates typical grid refinement of the lid-driven cavity flow case. Here the area of refinement is the upper left corner of the cavity, the main grid resolution is LDCF2 (101x101), and the sub-grid refinement factor is 4. The figure contains a Fieldview plot of the velocity vectors for the main grid along with the sub-grid velocity vectors. Also, there is a corresponding illustration of the mesh in Figure 4.15. By comparing the sub-grid results for velocity with the FLUENT results, the accuracy of the current implementation of grid refinement can be assessed.

Absolute error visualizations were performed to qualitatively evaluate the effectiveness of the sub-grid. A direct comparison of the absolute x-velocity error for the LDCF4 upper left sub-grid with a refinement factor of two (LDCF4-RF2-UL case) and the main grid LDCF4 case can be seen in Figures C.36 and C.37. The SBB boundary treatment is used for this visualization. This is the upper left corner of the cavity where the top of the cavity is moving to the right. In Figure C.36, there is a clear decrease in absolute x-velocity error when compared to the main grid x-velocity error (z-axes are scaled identically). This decrease in absolute error is representative for cases involving both the UR and UL corners and across boundary treatments.

The RMS error for the x and y velocities are graphed for various cases in Figures C.38 through C.41. All of the results are not presented here mainly because the general trends can be illustrated from a more limited set of results. The trends strongly indicate that the grid refinement does indeed reduce the error of the calculation in the area of the sub-grid. Figure C.38 graphs the x-velocity RMS error results for the LDCF1 (51x51) grid refined in the UL corner. This graph includes RMS error results for several different computations across all four boundary treatments (SBB, YU, BZ, and MDI). The first three sets of data which are labeled RF8, RF4, and RF2 come from the 3 sub-grids for this case with grid refinement factors of 8, 4, and 2. The fourth set of data labeled LDCF1 is the RMS error results for the entire LDCF1 grid taken from previous results in the boundary treatment studies. This is provided as a reference to the accuracy of the main grid calculation. The last set of data labeled SGA is the RMS error calculated from the nodes within the area of the LDCF1 main grid covered by the sub-grid. Thus, the SGA RMS error should be directly compared with the sub-grid RMS errors as it covers the exact same area. It is important to note that the SGA values are higher than the average values for the entire main grid because higher errors generally will occur in the area of grid refinement. Figure C.39 is a similarly structured graph displaying the x-velocity RMS error results for the LDCF4 (201x201) grid refined in the UL corner as well. Note that the z-axis is identically scaled in these two figures. Figures C.40 and C.41 displays y-velocity RMS error results for the LDCF1-UR case and the LDCF4-UR case and are structured similarly to Figures C.38 and C.39. Note that the z-axis is scaled identically for these two figures as well. The same trends can be seen in Figure C.40 as in the other two figures.

In these figures, there are several similar trends. First, the RMS error decreases as the grid refinement factor is increased (going from RF2 to RF8 for Figure C.38 or C.40). In addition, a substantial decrease is error is seen when comparing the SGA RMS error to the finest sub-grid. Thus, the accuracy of velocities is increased overall through the implementation of the sub-grid. Another interesting result is that in Figure C.38, the sub-grid RMS error for the finest sub-grid, RF8, is actually lower than the RMS error for the entire main grid LDCF1. This implies that the sub-grid is generating more accurate results then the entire main grid on average. This does not occur for the UR sub-grid as seen in Figure C.40, most likely because this is the region with the highest velocity errors. Lastly, the results across boundary treatments are relatively similar and show no distinct advantages.

To further quantify these results, Table C.11 lists the percent decrease in x-velocity RMS error going from the SGA RMS error to the sub-grid RMS errors for the UL cases with the same lattice spacing or sub-grid resolution. Table C.12 lists the percent decrease in y-velocity RMS error for the same grid resolutions, but for the UR cases. In Table C.11, the percent decrease in error hovers around between 60% and 80% regardless main grid resolutions and boundary treatments. The percent decrease is much less in the case of Table C.12, but this is most likely due to the particular area of grid refinement.

The next set of data presented involves the percent error. This data can help further qualify the effectiveness of the grid refinement. Tables C.13 and C.14 list the x-velocity percent error results for the sub-grid UL cases and SGA UL cases respectively. This data corresponds to the data presented in Table C.11. The next tables, Tables C.15 and C.16, list the y-velocity percent error results for the sub-grid UR cases and SGA UR cases respectively, which correspond to the data presented in Table C.12. These results reinforce the previous trends seen where percentage of nodes with lower percent error increases with a decrease in RMS error. Ideally, this would occur across all thresholds (<1% error, <5% error, and <10% error), but doesn't always hold true. In other words, a decrease in RMS error can happen through more nodes getting below the 10% error rather than seeing an increase in nodes with less than 1% error.

The last data presented for this section involves the refinement of the lower middle (LM) part of the cavity. This area of the cavity was chosen in order to test the consistency of the sub-grid solution with the main grid solution. There were no boundaries within these LM sub-grids, thus there was no opportunity for an increase in accuracy with the results. However, the results should verify that there is no degradation in accuracy when the sub-grid is implemented. The velocity RMS error results from the LM sub-grid were compared to the velocity RMS error results of the main grid for that same sub-grid area or the SGA RMS error. Table C.17 compares some of these results. The data shows that in nearly all of the LM cases, the velocity RMS error for the sub-grid was within 3% of the velocity RMS error of the main-grid for the same area. This supports that there is no significant degradation in accuracy with just the implementation of the sub-grid.

Grid Refinement for Flow in an Annulus

The grid refinement studies done for the case of flow in an annulus are structured similarly to those done for the lid-driven cavity flow case. Since the grid refinement is being performed around a stationary construct (See Figure 4.8), the geometry and flow parameters are identical to the CAS5 computations done previously for the constructs and annulus in the boundary treatment studies. The annulus resolutions used in these studies, ANN1, ANN2, and ANN3, will again be used in the grid refinement computations. Table 4.7 lists the multiple computations that will be done for each main grid resolution. The table shows that for the coarsest annulus main grid, ANN1, four computations are done for grid refinement factors 2, 4, 6, and 12. The second finest main grid resolution, ANN2, has three sub-grid computations done while the finest, ANN3, has two computations. Thus a set of nine sub-grid computations are done here to examine the effects of the grid refinement across main grid resolutions. Note that the finest sub-grid with lattice spacing $\Delta x = 0.0075$dm is four times finer than the finest main grid, ANN3, and occurs three times, once for each main grid.

Table 4.7 – Grid Refinement Computations for the 2D ANN Main Grid Resolutions

Grid	Main Grid Resolution	Lattice Spacing Main Grid Δx (dm)	Refinement Factor	Lattice Spacing Sub-Grid Δx (dm)
ANN1	41x41	0.090	2	0.0450
			4	0.0225
			6	0.0150
			12	0.0075
ANN2	81x81	0.045	2	0.0225
			3	0.0150
			6	0.0075
ANN3	121x121	0.030	2	0.0150
			4	0.0075

The previous boundary treatment studies involved the flow in an annulus around the circle (CIR), square (SQ), rectangle (RECT), and ellipse (EL) constructs. These four constructs will again be examined using grid refinement. The four boundary treatments were again tested to note any differences in accuracy. Therefore, 144 computations (4 constructs with 9 sub-grids using 4 boundary treatments) were performed for the CAS5 flow parameters. Since the CAS6 and CAS8 results were examined in the previous boundary treatment studies, additional computations are performed to examine the effects of increasing Reynolds number on sub-grids. For the CAS6 and CAS8 parameters, computations were performed for the most refined sub-grid for the ANN3 main grid. All constructs and boundary treatments were examined. Thus, 16 additional sub-grid computations (4 constructs using 4 boundary treatments) were performed for each of these flow parameters cases.

An additional set of computations were done in order to test altering the boundary treatment on the sub-grid with respect to the main grid. For the most refined ANN3 sub-grid and the ellipse construct, 12 additional computations were performed. These computations altered the sub-grid boundary treatment to differ from the main grid boundary treatment. In other words, for the ANN3 main grid computation done using the SBB boundary treatment, three additional sub-grids were computed using the MDI, BZ, and YU boundary treatments instead of the SBB. These computations are intended to highlight any sub-grid results that may differ based on the underlying main grid computation.

Figure 4.16 illustrates the grid refinement done in the annulus case. For this particular result, the area around a circular construct in the ANN2 (81x81) main grid resolution is refined by a factor of 3. The velocity vectors of both the main grid and sub-grid are shown along with the corresponding mesh used for the computation.

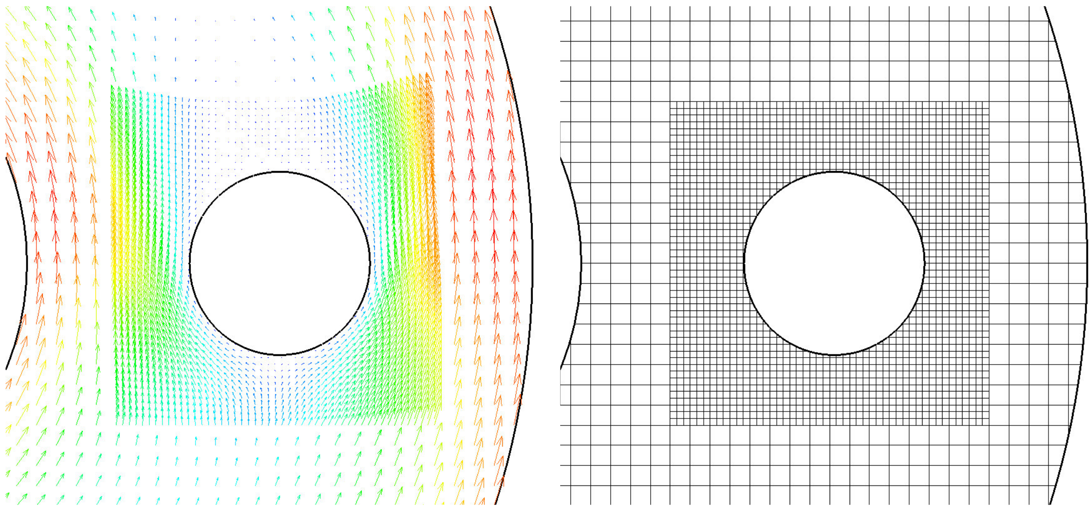


Figure 4.16 – Velocity Vectors and Mesh for Grid Refinement in the ANN Case

Absolute error visualizations were performed in order to initially compare the sub-grid with the main grid. Figure C.42 is a plot of the absolute y-velocity error for the ANN2-CIR sub-grid using a refinement factor of 3 (ANN2-CIR-RF3 case) using the MDI treatment. The maximum error for this case is 0.000212m/s. Figure C.43 is a plot of the absolute value of the y-velocity error for the ANN2-CIR main grid case using the MDI treatment. The maximum error near the construct surface is approximately 0.000594m/s. This illustrates that at first look, the sub-grid does reduce error for the flow surrounding the construct. However, further analysis of the RMS error and percent error quantifies the results.

The velocity RMS error results are in part presented in Figures C.44 to C.47. Since there is a large amount of data for these studies (144 computations), only a portion is presented here to illustrate the visible trends. As previously seen in the LDCF grid refinement studies, the grid refinement implementation for the annulus flow does reduce the error of the calculation when comparing SGA RMS error to sub-grid RMS error. Figures C.44 and C.45 are plots of the y-velocity RMS error for the ANN1-CIR case and the ANN3-CIR case, respectively. The graphs are similarly structured as the previous figures (See Figures C.38 to C.41) shown in the LDCF grid refinement studies. Note that the z-axes in the figures are scaled identically for easy comparison. Figures C.46 and C.47 are plots for the x-velocity RMS error for the ANN1-SQ case and the ANN3-SQ case. These plots are typical representations for x and y velocity RMS error plots across constructs.

Some of the same observations can be made about the RMS error trends seen in these plots compared with the LDCF grid refinement plots. In general a decrease in RMS error is seen when comparing the refined sub-grid errors to the SGA RMS error, or the RMS error calculated for the main grid on the area covered by the sub-grid. Also, the RMS error for the finest sub-grid does approach the average RMS error for the entire main grid in some cases. However, there is a distinction between boundary treatments that isn't present in the LDCF results. For the CIR and SQ construct, the SBB boundary treatment does relatively poorly compared to the other boundary treatments. This is most likely due to the SBB boundary treatment rough approximation of the surfaces.

The percent decrease in RMS error going from the SGA RMS error to the sub-grid RMS error is presented in Tables C.18 and C.19. Table C.18 lists the y-velocity RMS error for the most refined of the ANN-CIR sub-grid cases. The x-velocity RMS error for the most refined of the ANN-SQ sub-grid cases is presented in Table C.19. From the tables it can be seen that the percentage decrease in error is much greater for the coarser main grid ANN1 versus the more refined grid ANN3. This is expected because of the poor approximations of the coarser main grids, but also may indicate that the sub-grid effectiveness in terms of percentages is decreasing.

The percent error data for the ANN-CIR and ANN-SQ cases is presented for the most refined sub-grids in Tables C.20 to C.23. The percent error data for the sub-grids are presented first and subsequently in the next table the SGA percent error data is presented for comparison. The same general trends occur here as seen before in the LDCF cases with a higher percentage of nodes with low percent error occurring for the sub-grid versus the SGA main grid. This data also supports the reduced accuracy of the SBB boundary treatment with respect to the other treatments. This is primarily seen for the percent error <10% threshold.

The percent decrease and percent error results from examining an increase in the Reynolds number by using CAS6 and CAS8 flow parameters versus CAS5 flow parameters are presented for the ANN3-SQ-RF4 sub-grids in Tables C.24 to C.26. In examining the data, no boundary treatment distinguishes itself from the others with regards to its performance on sub-grids. The same absolute error trends are seen that were previously pointed out in the boundary treatment studies, with the CAS8 computations showing the higher sub-grid RMS errors and SGA RMS errors. In addition, the SBB boundary treatment underperforms in these studies compared to the interpolation boundary treatments. However, there is no clear interpolation boundary treatment that consistently performs better than the rest. This is also the case with the data for the other constructs.

The last set of data presented for the annulus grid refinement studies involves the mixing of boundary treatments for the ANN3-EL-RF4 sub-grid case. Twelve additional computations were done that mix the main grid boundary treatment and the sub-grid boundary treatment. Tables C.27 and C.28 list the percent decrease in RMS error for the x and y velocities for these cases. From the results, two interesting observations can be made. First, the accuracy of the sub-grid solution is mainly dependent on the accuracy of the main grid solution. The other observation is that boundary treatment has little effect on the final sub-grid RMS error except in the case of the SBB boundary treatment. The SBB treatment did worse than the interpolation methods across the boards with respect to RMS error.

Overall, the use of the lattice Boltzmann grid refinement in the annulus geometry brought more accurate velocity results when compared to the lattice Boltzmann calculations using only the main grid. There is a consistent decrease in velocity RMS error when the grid refinement is implemented. The greatest benefit is seen for the coarsest main grids when there is high velocity error occurring in the area of refinement. Other aspects of the grid refinement are examined further in the next set of studies.

4.3.3 Dynamics Studies

The numerical studies done in this section will test the accuracy of the current implementation techniques of the lattice Boltzmann method with respect to forces, shear stresses, and particle dynamics. The evaluation of these methods concerning forces and dynamics will also take into consideration the effects of boundary treatments and grid refinement. In the case of the force evaluation and shear stress evaluation, the computations involving the annulus geometry previously described will be used. The accuracy of these results will primarily be determined through comparisons with FLUENT data. Previous experimental and computational results will be compared to the dynamic studies involving moving boundaries.

Force Evaluation on a Stationary Construct

The force on a stationary construct will be evaluated using both the modified momentum exchange method listed in Equation 3.1 and the original method proposed by Aidun in Equation 2.50. The computations that will be used to evaluate the force have already been described in the previous sections involving boundary treatments and grid refinement. For the boundary treatment studies, 48 computations were performed on stationary constructs in an annulus under CAS5 flow parameter. For the grid refinement studies, 144 computations were performed under the same CAS5 flow parameters and annulus geometry. The differences between these computations were due to constructs, main grid resolution, sub-grid refinement, and boundary treatments. Additional grid refinement computations were performed for the ANN3-EL-RF4 sub-grid case where boundary treatments on the main grid and sub-grid were mixed and for the CAS6 and CAS8 most refined sub-grids for all constructs and boundary treatments. For this part of the dynamics study, the forces and moments are evaluated for each computation using the two methods and compared to the FLUENT results for force. The constructs under consideration are the circle (CIR), square (SQ), rectangle (RECT), and ellipse (EL) constructs. The forces and moments under consideration are the x-direction force, y-direction force, and torque (positive clockwise).

The comparison between the modified momentum exchange method in Equation 3.1 and the Aidun method in Equation 2.50 for force evaluation was intended to test the methods across the four boundary treatments. However, the results clearly showed that the only boundary treatment that the Aidun method gave accurate force results for was the SBB treatment. Conversely, the modified momentum exchange method gave accurate results for the interpolation boundary treatments, but not for the SBB treatment. This makes sense because the Aidun method was based on SBB, while the modified momentum exchange method is based on interpolation. The consequence of this is that the force results presented here for different boundary treatments are consistent with the force evaluation method that works. In other words, the SBB force results used the Aidun method while the interpolation results used the modified momentum exchange method.

Tables C.29 to C.32 list the force and moment results for the four constructs for the main grid calculations. The percent error is calculated for the lattice Boltzmann results assuming that the FLUENT force and moment results are the accurate solution. From the tables it is clear that the lattice Boltzmann implementation does better as resolution increases from ANN1 to ANN3, but significant error still exists at the highest main grid resolution. It is also interesting that the force in the y-direction is relatively close to the FLUENT in contrast to the x-direction force and torque. The difference in force and moment error when comparing error across constructs is more pronounced than differences in RMS velocity error across constructs seen in previous studies. For example, the x-direction force error for the RECT construct is 200% or more for the ANN3 main grid while the error for the EL construct for the same grid is generally under

20%. Lastly, there does not seem to be a clear better boundary treatment performer since the force and moment errors seem to be high across the board.

The force and moment error for various sub-grid cases are presented in Tables C.33 to C.36. For the sub-grid results, there is a clear improvement in error over the main grid results especially for the ANN3-RF4 cases. Overall, the error in the x-direction force and z-torque are decreased for all constructs. The y-direction force error is also decreased in nearly all cases. The ANN3-RF4 cases are the most accurate results overall and generally show errors less than 5%. While for different cases a different boundary treatment may give the best result for the force or moment, the performance of the MDI boundary treatment is the most consistent across all cases. The force and moment error when using the MDI boundary treatment are consistently lower when you take all three force and moment measurements into consideration. In contrast, the SBB boundary treatment is more likely to have higher force and moment errors. However, considering the high velocity RMS errors, the SBB boundary treatment does well at these low resolutions.

The last sets of force and moment data presented include data for the flow parameter studies in Table C.37 and data for the mixed boundary treatment studies in Table C.38. The data in Table C.37 is results for the ANN3-RF4-EL sub-grid for the CAS5, CAS6, and CAS8 flow parameters. The RMS velocity error for CAS8 studies is known to be higher than CAS5 and CAS6, therefore an increase in force and moment error is expected as well. Surprisingly, this is not the case for the x-direction force, and is only the case for the y-direction force and some of the z-torque moments. In contrast, CAS5 and CAS6 force and moment errors are comparable. The results for the boundary treatment mix cases in Table C.38 show that although velocity RMS errors may be comparable for the interpolation boundary treatments (See Tables C.27 and C.28), the force results do not match the same trend. In fact, the MDI and SBB boundary treatments seem to perform best on the sub-grid for all the main grid boundary treatment cases.

The force and moment results presented here indicate that the Aidun method or the modified momentum exchange method for force evaluation used in the lattice Boltzmann gives results within reason compared with FLUENT. The majority of results using the lattice Boltzmann refined grids are within 5% of the FLUENT results. This validates the use of the lattice Boltzmann force and moment evaluation methods.

Shear Stress Evaluation for Flow in an Annulus

The accuracy of calculations for the 2D shear stress and shear rate, σ_{xy} and S, (See Equations 3.26 and 3.32) depend on the accuracy of the velocities and velocity gradients calculated. Therefore, an increase in velocity accuracy due to boundary treatments or grid refinement should lend to greater accuracy in shear stresses calculated within the fluid. The computations that will be used for analyzing the shear stresses in the fluid will be the same ones used for the force evaluation section because the shear stresses around the construct are of primary concern. Again, these are the CAS5 annulus computations for flow around a stationary construct, where four constructs, four boundary treatments, and different grid resolutions were explored. The lattice Boltzmann data is compared to FLUENT data for shear stress in the fluid and the shear rate for the same geometry and flow parameters. A custom calculation was used in FLUENT to obtain the shear stress data with velocity gradients already calculated by FLUENT. The shear rate is always calculated by FLUENT. It is important to note that while FLUENT can calculate the wall shear stress at the construct wall; the same calculation is not done in the lattice Boltzmann computations. Only the shear stress in the fluid and the shear rate is calculated by the lattice Boltzmann. However, through some processing of the lattice Boltzmann velocity data (which methods are not being explored or analyzed in this study) the wall shear stress is obtainable by calculating velocity gradients normal to the wall. It is therefore inferred that if the shear stress in the fluid and shear rate are accurate for the lattice Boltzmann method, then calculated wall shear stresses would also be accurate depending on the quality of the wall shear stress calculation.

A more detailed examination of the data can be performed by plotting the shear stress and shear rate across the annulus at the construct. The difference in accuracy between the main grid versus the sub-grid, between different sub-grid resolutions, and between boundary treatments is of primary interest. The sub-grid should produce an increase in accuracy with an increase in refinement for these shear calculations since the velocities were shown to be more accurate in the grid refinement studies. Figure 4.17 illustrates the position of the plots for this section of the analysis. The plots occur at Y = 1.8dm across the annulus and cut across the various constructs approximately between X = 2.8dm and X = 3.2dm.

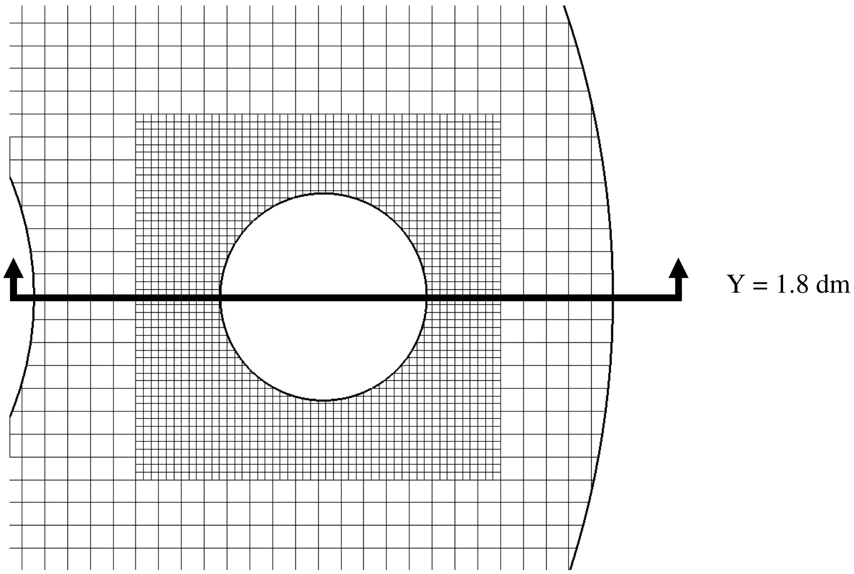


Figure 4.17 – Illustration of Line Plot at Y = 1.8dm

The lattice Boltzmann results for shear stress and shear rate in the fluid compare well with the FLUENT results. It should be noted that the calculation methods previously described, the lattice Boltzmann and finite difference methods (Appendix B and Equations 3.26 to 3.31), yielded identical results and are therefore just referred to as the lattice Boltzmann results. Qualitatively, contour plots for the shear stress and shear rate match well for all boundary treatments across all constructs. Figures C.48 and C.49, illustrate this for the annulus geometry with a circle construct and ellipse construct respectively. Figure C.48 shows contour plots of the FLUENT (top) and lattice Boltzmann (bottom) results for shear stress around the circular construct. Note that the smoothness of the contours is improved within the boundary of the refined grid (black box border) for the lattice Boltzmann results. The main grid resolution was at the ANN2 (81x81) resolution while the refinement factor was 6. The units for the shear stress are in dN/dm^2 (which equals 0.1Pa). Figure C.49 shows contour plots of the FLUENT (top) and lattice Boltzmann (bottom) results for shear rate around an elliptical construct. Note again that with grid refinement, the contours are much smoother compared to the coarser main grid computation. Here the main grid resolution was at the ANN3 (121x121) resolution while the refinement factor was 4. The units for the shear rate are 1/s.

Figures C.50 and C.51 are plots of the shear stress and shear rate across the ANN2 annulus at the CIR construct along Y = 1.8dm. While the BZ boundary treatment is being used here, these results for main grid to sub-grid comparisons are representative of what is seen across boundary treatments. In the figures, the solid line represents FLUENT results, while the open squares represent the main grid results. It is evident in both plots that while the main grid results track well with the FLUENT results, there is definitely room for improvement. The open triangles represent the sub-grid results, where the grid refinement factor was set to 6. It is clear from the plots that the sub-grid values track much closer to the FLUENT values than the main grid values. This illustrates the sub-grid's usefulness in enabling better approximations of shear stress or shear rate closer to the construct. This also validates the velocity results for the grid refinement and further endorses the accuracy of the method overall.

To further examine the accuracy of the grid refinement as the refinement factor is increased, sub-grid values for shear stress and shear rate are plotted in Figures C.52 to C.53. In Figures C.52 and C.53, results from grid refinement factors 2, 3, and 6 are plotted against each other for the ANN2 annulus near the right edge of the RECT construct along Y = 1.8dm. The open squares represent the lowest grid refinement of 2, the open triangles represents grid refinement of 3, while the thin line represents the highest refinement. The thicker line is the FLUENT solution. From both figures, it is clear to see that as the sub-grid is refined, the lattice Boltzmann results begin to track the FLUENT results more closely. These results were done with the BZ boundary treatment; however this trend across refinement factors is seen for the shear stress and shear rate regardless of boundary treatment.

Another interesting comparison that can be done is comparing the results for sub-grids with equal lattice spacing (Δx) values that resulted from different main grid resolutions. Figures C.54 and C.55 are shear stress and shear rate plots for the same situation at the right edge of the RECT construct using the MDI boundary treatment, but plotting results from both the ANN2 and ANN3 cases. In the plot, the triangles (open is ANN2, filled is ANN3) represent results from sub-grids with the same resolution or lattice spacing (See Table 4.7). The lines (dashed is ANN2, solid is ANN3), also represent results from sub-grids with the same lattice spacing. From the plots it is clear that although the sub-grids may share the same resolution and lattice spacing, the accuracy of the results is not necessarily comparable. This is most likely due to the difference in information being passed into the sub-grids from the difference in main grid resolution. Logically, the more accurate results should come from the sub-grid being passed information from the more accurate main grid. This seems to be the case as the ANN3 results track more closely to the FLUENT solution than the ANN2 results.

While trends concerning grid refinement may be common across boundary treatments, there is a difference in the accuracy between boundary treatments that must be examined. The sub-grid values for shear stress and shear rate are plotted in Figures C.56 and C.57 for the four boundary treatments (SBB, MDI, BZ and YU) for the ANN3 annulus at the left edge of the EL construct for a grid refinement factor of 4. Both figures clearly show that the SBB treatment does worse compared to the other boundary treatments. The YU treatment is second worse while the BZ and MDI treatments perform almost identically. This is somewhat to be expected as it correlates with the RMS error results previously presented.

In general, the results for shear stress and shear rate match well with the FLUENT results. The most divergence between lattice Boltzmann and FLUENT results is seen at the surface of the construct, in cases where the flow is more complex due to geometry. While the CIR and SQ constructs had relatively linear shear stress and shear rate plots, the RECT and EL constructs had more areas of rapid change at the surface. This is the primary reason those cases were presented as they brought out contrasting results more readily. Even in these cases, the shear stress and shear rate results of the lattice Boltzmann for refined grids is remarkably close to the FLUENT results and validate that the velocity solution of the lattice Boltzmann. In addition, the data suggests that the shear results from the lattice Boltzmann are comparable to results from FLUENT.

Lubrication Force between Two Approaching Cylinders

The first dynamic case measures the forces calculated by the lattice Boltzmann when two circular cylinders approach each other in a channel. This problem was previously performed by Ding & Aidun (2003) to test the implementation of their near contact particle lubrication forces. In that study, the dimensions of the channel were 2L x L and the diameter of the cylinders was held constant at 0.575L (See Figure 4.9). The value of L was varied from 32 (a coarse grid) to 1024 (a very fine grid) where $\Delta x = \Delta y = 1$. The velocity of the cylinders was held constant at $U/2 = 0.01$. The kinematic viscosity for the coarsest grid was set to 1/8 and was increased for the finer grids to maintain the same Reynolds number. Note that all of these parameters are non-dimensional. The non-dimensional force ($F/\upsilon\rho U$) calculated by the coarser grids was then compared to the finest grid calculation. In this study, the same non-dimensional parameters are used, except that the finest grid calculation was done for L = 512. Table 4.8 shows the non-dimensional parameters for the five cases computed. Computations are performed for the SBB and BZ boundary treatments. In addition, results were found for the non-dimensional force calculated implementing lubrication and force calculated without implementing lubrication. Thus for this part of the study 20 computations (2 grid resolutions, 2 boundary treatments, and 2 cases for force calculation) were performed. In addition to these primary 20 computations, 3 sub-grid computations are computed for the SBB boundary treatment for the L = 64 case. The refinement factors for these computations were 2, 4, and 8.

Table 4.8 – Lattice Boltzmann Parameters for the 2D Lubrication Grids

Resolution	Lattice Spacing Δx	Channel Width L	Construct Diameter d	Kinematic Viscosity υ
65x32	1	32	18.4	1/8
129x64		64	36.8	1/4
257x128		128	73.6	1/2
513x256		256	147.2	1
1025x512		512	294.4	2

The calculated lubrication force in this study for squeeze film lubrication for the 2D cylinders is based on the analytical value. The analytic solution was based on the differential force generated when squeezing fluid between two plates, but was integrated for the two circular cylinders problem. Equation 4.5 shows the formula used to calculate the lubrication force $\mathbf{F}_l$ between two approaching cylinders per unit depth l. The kinematic viscosity υ, fluid density ρ, relative approaching velocity **U**, cylinder radius R, and gap size ε are all parameters that appear in Equation 4.5. This lubrication force is used when it becomes larger than the calculated force between the two particles.

$$\frac{\mathbf{F}_l}{l} = 3\upsilon\rho\mathbf{U}\frac{R}{\varepsilon}\left[1+\left(\frac{R}{\varepsilon}\right)^{1/2}\arctan\left[\left(\frac{R}{\varepsilon}\right)^{1/2}\right]\right] \tag{4.5}$$

Figures C.58 and C.59 show the non-dimensional force F = $\mathbf{F}_l/l\upsilon\rho\mathbf{U}$ versus the non-dimensional gap size S (where S is gap size ε divided by the radius R) for computations using the SBB and BZ boundary treatment respectively on the various grid resolutions. The analytical solution is also plotted on these figures as well. The results for the different boundary treatments were essentially similar. As a reference, Figure C.60 shows forces calculated without using the lubrication calculation. While the forces do show a better approximation as the resolution increases, it is still necessary to implement the lubrication correction for the finer grids. The results from the three sub-grid calculations show that the sub-grid approach calculates the identical forces that the equivalent main grid calculates. Note that this is done for the case of no lubrication forces added. The results are shown for one case in Figure C.61. The forces calculated for the 129x64 main grid and the 129x64-RF8 sub-grid are plotted against the 1025x512 main grid calculated forces. Note that the sub-grid and finer main grid have equivalent resolutions. The figure shows that the forces calculated by the sub-grid are virtually identical to those calculated by the equivalent resolution main grid.

The results from these lubrication cases show that the current implementation allows for the correct force to be used when particles are in close contact. In addition, the results from the grid refinement studies done here show that the sub-grid does calculate forces for moving boundaries identically to an equivalent resolution main grid.

Particle Sedimentation in a Channel

For the second dynamic case, the sedimentation of a circular cylinder in a long vertical channel is modeled. The problem was previously done experimentally by White (1946) where thin wires were dropped in a viscous fluid and the terminal velocity was measured. The ratio of the channel width (L/d) to the wire diameter was varied. The results for the dimensionless quantity $C = D/\mu U_t$ were plotted against the ratio L/d and a best fit line was determined. D is the drag force per unit length of cylinder, μ is the dynamic viscosity and U_t is the terminal velocity. The same problem was solved analytically by Faxen (1946) and Takaisi (1955). Equations 4.6, 4.7, and 4.8 are the equations for the experimental and analytical solutions of White, Takaisi, and Faxen respectively. Equation 4.7 only gives the first few terms of the total formula given by Faxen.

$$\frac{\mathrm{D}}{\mu \mathrm{U}_t} = \frac{6.4}{\log(L/d)} \tag{4.6}$$

$$\frac{\mathrm{D}}{\mu \mathrm{U}_t} = \frac{4\pi}{\ln(L/d) - 0.9156} \tag{4.7}$$

$$\frac{\mathrm{D}}{\mu \mathrm{U}_t} = \frac{4\pi}{\ln(L/d) - 0.9157 + 1.7244(L/d)^2 - 1.7302(L/d)^4} \tag{4.8}$$

For this study, several grids of varying resolutions are used to cover a similar range of L/d values used by White. Table 4.9 lists the grid resolutions used and the values of various parameters. Note that the number of lattice spacing across the construct diameter is kept constant (~10 lattice spacings) for all cases. The density of the fluid was set to $1kg/dm^3 = 1000kg/m^3$ and the kinematic viscosity is set at $0.001dm^2/s = 0.00001m^2/s$. This in turn set the dynamic viscosity μ to 0.001kg/dm·s. The length of the channel is approximately 50d which means the construct is placed approximately 25d from the inlet and outlet of the channel. The boundary conditions tested for this study are the closed cavity condition, where the SBB boundary treatment is applied at the inlet and outlet, and the free boundary condition or the Stokes boundary condition, which was explained in Chapter 3. Since these conditions will not precisely mimic an infinite fluid, the results should fall in between the analytic solution and the experimental results. Three boundary treatments (SBB, BZ, and YU) were also tested for each resolution. Thus, 24 computations (4 grid resolutions using 2 boundary conditions and 3 boundary treatments) were performed for the initial part of this study.

Table 4.9 – Lattice Boltzmann Parameters for the 2D VCF Grids

Resolution	Lattice Spacing Δx (dm)	Channel Width L (dm)	Construct Diameter d (dm)	L/d
51x501	0.03	1.53	0.30600	5
101x501		3.03	0.30300	10
201x501		6.03	0.30150	20
301x501		9.03	0.30100	30

To examine the effects of further increasing grid resolution on the motion of the particle, grid refinement was applied to the moving particle. An additional 24 computations were done using the Stokes boundary condition and grid refinement, with a refinement factor of 2 and 4, for the four grid resolutions and three boundary treatments. For a refinement factor of 2, the lattice spacing decreased to $\Delta x = 0.015$dm and effectively increased the number of lattice spacings across the construct by a factor of 2. Similarly, this occurred for a refinement factor of 4. The differences in particle forces, velocities, and terminal velocities between simulations using different boundary conditions, boundary treatments, and grid resolutions are illustrated in the results presented.

Figures C.62 to C.68 summarize the results for the 2D computations for particle sedimentation. Figures C.62 and C.63 plot the dimensionless drag constant C versus the grid parameter L/d for the simulations using the Stokes boundary condition and the cavity boundary condition, respectively. There is little to no significant difference in these 2D results as they are almost identical across all boundary treatments. Thus, the Stokes boundary condition was used for the rest of the 2D and 3D simulations involving particle sedimentation. However, from these figures it is clear that the particle terminal velocity does not fall between the analytic solutions of Faxen and Takaisi and the experimental results of White as expected. Figure C.64 illustrates the drag constant results using the BZ boundary treatment with and without grid refinement. When grid refinement is applied, the drag constant C does fall between the analytic and experimental results. This occurs for both refinement factors used. This infers that the grid resolution across the diameter of the construct for just the main grid (~10 lattice spacings) is insufficient to produce accurate results, and when grid refinement is applied, the terminal velocity falls more in line with both analytic and experimental results.

The particle velocity and forces are shown in Figures C.65 and C.66 for the VCF3 grid. Figure C.65 illustrates the differences in particle velocities as it sediments under different boundary treatments. All three boundary treatment exhibit velocity variations while sedimentation occurs. The SBB boundary treatment shows dips in velocity while conversely the YU boundary treatment exhibits small increases in velocity. In contrast, the BZ boundary treatment shows an oscillating behavior. Figure C.66 is a plot of the calculated particle force in the sedimentation direction versus the distance traveled. The plot shows that the most variation in the force is seen with the SBB boundary treatment. The YU boundary treatment shows a smaller range of force values and BZ boundary treatments show even less variation. Finally, Figures C.67 and C.68 compare the particle velocity and force for the VCF3 grid refined cases using the BZ boundary treatment. The velocity plot in Figure C.67 clearly shows a higher terminal velocity being reached when grid refinement is implemented. Less oscillation in velocity is also observed. For the forces plotted in Figure C.68, less overall variation is seen with grid refinement.

Two Particle Sedimentation in a Channel

For the third dynamic case, the sedimentation of two circular cylinders in a long channel is performed. Feng et al. (1994) and Aidun & Ding (2003) have published work simulating a similar problem. Since the Aidun & Ding study used lattice Boltzmann as their computation technique, this study will replicate the same non-dimensional parameters used in that study. In this study, the channel width is four times the diameter of the particles (See Figure 4.11). The diameter of the particles is set to 32 where $\Delta x = \Delta y = 1$. Thus the resolution across the channel is 128 lattice units. The length of the channel is set to approximately 100d, while the kinematic viscosity is set to ¼. The channel length in this study is different than that of Aidun & Ding since different boundary conditions are used at the inlet and outlet. In the study by Aidun & Ding, a net body force was specified and varied to drive the flow. In this study the same type of net body force is imposed in order to cause the sedimentation. The net body force is varied in this study as well in order to observe changes in the particle interaction. Also, the Aidun & Ding study used the SBB boundary treatment for all computations, while in this study both the SBB and BZ boundary treatments are used and compared.

Figure C.69 shows the typical particle path results plotted versus non-dimensional time for the sedimentation of two particles in a channel for the SBB boundary treatment. For this case, the non-dimensional net body force on the particles is set to 0.305 and the Reynolds number was 2.96. The interaction between the particles as they sediment is similar to the results of Feng et al. (1994) and Aidun & Ding (2003). Initially, the second particle trails the first particle and both drift towards the right wall. Following this, the particles come in close contact where lubrication forces are significant and the trailing particle ends up drifting towards the left wall. The particles then enter a damped oscillation. Figure C.70 is a comparison of the SBB boundary treatment result with the BZ boundary treatment results. For the BZ calculation, the non-dimensional net body force was set to 0.300 and the Reynolds number was 2.93. The figure shows that although the Reynolds number are similar for both calculations, the result is differs. The position of the particles is similar for the initial near contact, but then differs as the computation proceeds. Notably, the amplitude of the BZ boundary treatment trailing particle is a bit larger than the SBB boundary treatment trailing particle. Also, the period of the motion is different. This is evidence of the sensitivity of the system to boundary treatments.

The period doubling behavior seen by Aidun & Ding (2003) occurred as the net body force applied was increased up to a Reynolds number of 4.216. For the current studies, the period doubling behavior was not observed as the Reynolds number increased. This is partly due to the sensitivity to lattice Boltzmann implementation of the two particle sedimentation system whereby the correct net body force to apply was unknown. Finding the correct net body force to produce doubling could take a significant of time as each calculation takes several days to complete. The differences between lattice Boltzmann implementations also includes handling of the particle motion and boundary conditions. Specifically for boundary conditions, Aidun & Ding (2003) modeled an infinite channel by adding fluid to the inlet and removing fluid from the outlet during the particle sedimentation. In this study, the channel was a long but finite channel. This difference affected the results of the current study especially as the particles approached the bottom of the channel. It is possible that the end of the finite channel was reached before the behavior could be observed. These small factors can have a cumulative effect on the results. As seen in Figure C.70, the boundary treatment itself can cause a difference in outcomes. Despite this, the current implementation does give the correct qualitative behavior as seen in previous studies of the two particle sedimentation problem.

Figure C.71 is a representative result of the behavior as the Reynolds number was increased for the SBB boundary treatment. As the net body force was increased to 0.324, the Reynolds number increased to a value of 3.289. This is still within the region of damped oscillations as seen by Aidun & Ding (2003). However, as the net body force was increased to 0.340, the particle interaction became more complex as seen in the figure. The maximum Reynolds number seen for this case was 7.263, while an average Reynolds number was 4.267. The average Reynolds number would indicate this is the range of the periodic doubling behavior; however the phase space plot doesn't indicate periodic doubling. This may be due to the finite length of the channel and the calculation not reaching a periodic state and remaining in a state where transient behaviors are still present. However, the behavior seen here is representative of the upper period branch, but the computation was not carried out long enough to reproduce its periodic nature.

Only a definitive reproduction of the lower stable periodic branch solution is seen using the finite channel. This is due to not matching boundary conditions and the time constraint of not being able to run many values of the net body force. Further results, not included here, using the Aidun boundary condition for the channel have shown periodic solutions with behavior similar to the upper branch solutions and period doubling. Further inspection across a range of net body forces is needed to reproduce the period doubling behavior. These results support the soundness of the current implementation.

Motion of a Neutrally-Buoyant Particle in an Annulus

The motion of a neutrally-buoyant particle is examined in the annulus geometry. Previous computations done for the boundary treatment studies and the grid refinement studies are used as a guideline. The geometry of the previous annulus studies where the circular construct was held stationary is used. The flow parameters are varied from the previous annulus studies. The inner and outer wall velocities are set in order to induce a rigid body rotation case. Thus the outer wall linear velocity is set at 0.0370dm/s and the inner wall linear velocity is set at 0.0138dm/s for the 2D case. The kinematic viscosity is set to 0.25dm^2/s. A steady solution is first calculated for the fixed construct case. Starting from the steady solution, the construct is then no longer restricted and is allowed to freely move within the annulus. The dynamic motion is observed following its release.

The motion of a circular cylinder particle is presented here for both main grid calculations and moving refined grid calculations. This contrasts the use of force calculations from a rough main grid versus the calculations from a finer sub-grid. An illustration of the moving construct-centered sub-grid for the ANN2-RF6 case is shown in Figure 4.18. After the particle is released from the rest initial position, it travels in a circular orbit within the annulus. The red arrows indicate the direction of the wall movement. As the particle center moves across the main grid, the sub-grid shifts or "floats" along with the particle. In the figure, the velocity vectors are colored by velocity magnitude. The lattice spacing for the ANN2 main grid is $\Delta x = 0.045$dm. A refinement factor of 6 is used for the moving sub-grid, which corresponds to a lattice spacing of $\Delta x = 0.0075$dm.

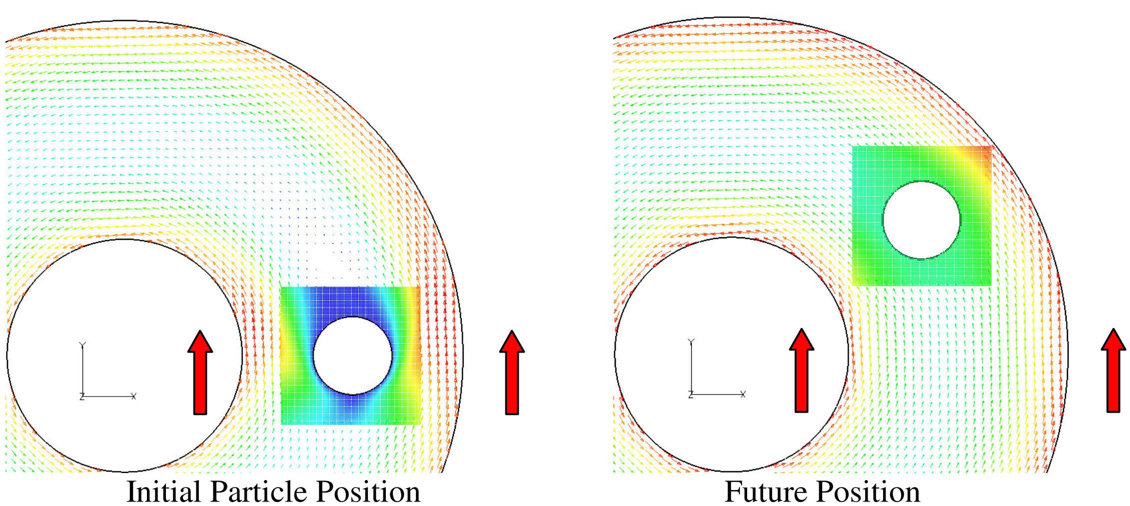


Figure 4.18 – Illustration of a Moving Sub-Grid in the Annulus Geometry

Figure C.72 is the particle path for the CIR construct when released from its stationary position. These results are for the ANN3 grid which has a resolution of 121x121 grid nodes. The three boundary treatments tested were the SBB, BZ, and YU boundary treatments. The plot shows that the particle follows a circular orbit when released for all the boundary treatments. This corresponds to the rigid body rotation induced by the wall velocities. In comparing to experimental results for similar studies, lighter than water microcarriers in the study by Botchwey et al. (2000) tended to move to an equilibrium radial position in a rotating cylinder. Also, in a study by Wereley and Lueptow (1999) that examined the motion of particles in Taylor Couette flow, particles were found to follow streamlines closely, but not exactly due to the curvature of the velocity field near the particle. Thus the results here are similar to those seen in experiments.

The grid refinement was applied to this case for a very coarse grid with resolution 21x21grid nodes (termed ANN0). This corresponds to a lattice spacing of $\Delta x = 0.18$dm. For the SBB boundary treatment, the computation on the main grid was unstable as soon as the construct was released. This was due to the limited number of nodes that resolved the construct area (the diameter of the construct was approximately 2 lattice spacings wide). When a moving sub-grid of refinement factor 6 was applied, the computation remained stable. In addition, the construct did return to the original position as expected. The lattice spacing of sub-grid was $\Delta x = 0.03$dm, which is equivalent to the ANN3 main grid. In Figure C.73, the two particle paths of these two cases are plotted.

Overall for the 2D dynamics studies, the lattice Boltzmann performs well with respect to predicting forces and shear stresses on a construct. It also performs as expected with the particle sedimentation studies and with the studies involving the neutrally buoyant particle in the annulus. The dynamics studies also further reinforced the use of the grid refinement in bring about more accurate results. Based on these studies, the 2D lattice Boltzmann model implemented here should give reasonable results for bioreactor simulations.

4.3.4 Simulations

Simulations of the RWV bioreactor are performed in 2D using the annulus geometry and the rectangular construct. While the actual geometry and flow within the RWV bioreactor is 3D in nature, a 2D simulation was explored nonetheless. Thus, the flow patterns seen in the 2D simulation may not accurately reflect the actual 3D flow. The physical parameters of the RWV bioreactor model used by Neitzel et al. (1998) have been closely followed. However, the viscosity of the fluid could not be reproduced exactly due to instability as the wall velocities of the annulus were increased to match the experimental and computational studies. Table 4.10 shows a comparison of the parameters used for the simulation presented here and parameters in the study by Neitzel et al.

Table 4.10 – Lattice Boltzmann Parameters for the 2D Simulation

Parameter	LB Simulation	Neitzel et al. (1998)
Inner Wall Radius	0.1015dm	0.1dm
Outer Wall Radius	0.292dm	0.29dm
Inner Wall Velocity	0.4dm/s ~ 13 RPM	Varied (E) 13 RPM (C)
Outer Wall Velocity	0.4dm/s ~ 37 RPM	Varied (E) 37 RPM (C)
Construct Radius	0.03dm	0.02875-0.03285dm
Fluid Kinematic Viscosity	$0.0005 dm^2/s$	$0.0001 dm^2/s$ = 1cS
Fluid Density	$0.816 kg/dm^3$	$0.816 g/cm^3$
Construct Density	$0.830 kg/dm^3$	$0.830 g/cm^3$

E – Experiments C – Computations

From Table 4.10, most of the annulus and construct geometry were matched. The lattice used in the simulation is similar to the ANN2 lattice used in the previous studies except it was scaled down by a factor of six. The lattice spacing was set to $\Delta x = 0.0075$dm for the main grid resolution of 81x81 lattice nodes. A sub-grid using a refinement factor of six is also implemented to ensure more accurate construct motion. The relaxation constant corresponding to the desired kinematic viscosity was $\tau = 0.7$. The 2D construct used here was the rectangular construct and it was initially placed at the 3 o'clock position. A steady state solution for the flow around the construct was first generated. Then the construct was released and its motion was observed. In the experimental studies done by Neitzel et al., the rotation speeds were varied in order to observe 3 distinct construct motions. The construct stayed stationary near the inner wall, stayed stationary near the outer wall, or moved in a small orbit about a fixed point. In the 2D lattice Boltzmann simulations, a force applied to the construct, which simulates a net body force as done in the two particle sedimentation cases, will be varied in an attempt to reproduce the construct motion seen experimentally. This is similar to changing the density of the construct.

The results of the 2D simulations are presented for three cases. The first is the case where the motion of the construct is characterized by staying stationary in the flow. The second case is the case where the construct orbits about the annulus and the last is the case where the construct oscillates about a fixed point. Figure C.74 is a visualization of the flow around the rectangular construct using velocity vectors after it has been released and is floating at a stationary point. Note that the vectors are colored by velocity magnitude and the scale is in dm/s. The 2D flow around the construct is different from the 3D flow seen in the experiments and previous computations. The primary difference is seen in that the wake structure. The 2D wake structure contains 2 vortices, while the 3D wake structure seen in the experiments only contains one area of recirculation.

Figures C.75 to C.77 are the construct paths plotted for three cases mentioned. In Figure C.75, the construct essentially stays stationary in the area where it was released. For this case, the net body force applied to the construct was approximately equal to the drag force calculated for a construct held stationary. The construct path for the orbiting case is presented in Figure C.76. In this case the net body force applied to the construct was around 20% of the stationary drag force. For the last case presented, the construct oscillates about a point. This construct path is plotted in Figure C.77. The red arrow shows the motion of the construct. The oscillating motion is up and down, rather than the circular motion seen in experiments. However, qualitatively the behavior is similar, considering that these are 2D computations and the wall velocities were not varied as in the experiments.

The last figure presented is a contour plot of the x-y shear stress for the flow around the floating rectangle construct. This plot is presented in Figure C.78. From the legend, the maximum positive value for the shear stress calculated in the fluid is approximately 0.01834dN/dm^2, while the maximum negative value for the shear stress is around 0.02750dN/dm^2. This translates to 1.834 dynes/cm^2 and 2.750 dynes/cm^2. These are definitely in the range of reasonable shear stresses measured in previous bioreactor experiments. Another way to estimate the 2D shear stress is to divide the drag force by the surface area of the construct. In the case of the rectangular construct, the drag force for the stationary case construct is 0.004024dN, while the surface area is 0.204dm^2. The resulting estimated shear stress is 0.019725dN/dm^2 or 1.97 dynes/cm^2. As noted in the literature review in Chapter 2, average shear stress on a one week old construct was estimated at 1.5 dynes/cm^2 (Freed & Vunjak-Novakovic 1995). The computational results in the study by Neitzel et al. showed shear stresses on the order of 0.8 dynes/cm^2. Thus, the simulations yield a reasonable value for the shear stress compared to previous results.

Overall, the 2D simulations have shown good qualitative results when compared to previous published results. The behavior of the floating rectangle construct in the simulated RWV bioreactor is similar to the behavior of constructs in the experimental setup. A stationary and orbiting regime were found for the 2D simulations. In addition, the shear stress calculated in the fluid by the lattice Boltzmann simulation is on the order of previously published results. The 3D simulation results, however, should serve as a better indication of the performance of the lattice Boltzmann for the RWV bioreactor problem.

4.4 3D Results

The 3D lattice Boltzmann code is based on the D3Q19 model. Most of the methods implemented in the 2D model as described in Chapter 3 are implemented in the same way for the 3D model. This reflects the flexibility of the methods used in this study. Since this is a different lattice Boltzmann model than the one used for 2D, the accuracy of the model using these methods should also be characterized. The 3D results presented here are from numerical studies that closely follow the numerical studies done with the 2D lattice Boltzmann code to assess boundary treatments, grid refinement, and dynamics. Because of memory limitations in running the 3D implementation, the exact resolutions of the 3D studies will not mirror those of the 2D study. However a range of resolutions is undertaken. Full 3D simulations of the RWV bioreactor are also done and the results are reported after the results from the numerical studies. Appendix D contains the results of the numerical studies and simulations performed using the 3D lattice Boltzmann code. This appendix is referenced throughout the presentation of the 3D results.

4.4.1 Boundary Treatment Studies

Similar to the boundary treatment studies done in 2D, the 3D studies presented here compare the performance of the shifted bounce-back (SBB), Bouzidi quadratic (BZ), and Yu linear (YU) boundary treatments. All of these treatments can be easily extended to 3D, but the question remains if they perform equally as well in 3D. The 2D results point to the SBB boundary treatment results having the most error while the MDI and BZ treatments perform very similarly. The 3D results will assess if the same boundary treatment behavior observed using a 2D lattice Boltzmann model holds true for the 3D lattice Boltzmann model. However, there is one difficulty in extending the MDI treatment to 3D and that is the lengthy computation time for 3D interpolation using MATLAB. Because of the unreasonable amount of time that a full MDI 3D simulation takes, the testing of the MDI boundary treatment will be extremely limited in 3D. The MDI boundary treatment will only be tested in the shifted channel cases. While this is not ideal, it will allow for a comparison of the boundary treatments. The numerical results presented are intended to characterize the performance of these treatments in 3D.

Body Force Driven flow with shifted boundaries

The flow in a 3D square channel due to a pressure gradient was simulated using the lattice Boltzmann code by driving the flow with a body force. The boundary distance Δ from the computational nodes to the square channel was varied for five cases (See Figure 4.2). As with the 2D study, the initial case has $\Delta = 0.5$, and only the shifted bounce-back (SBB) boundary treatment was tested. This is due to the fact that the other three boundary treatments should collapse to the SBB boundary treatment when $\Delta = 0.5$. The next four cases, D1, D2, D3, and D4, correspond to boundary distances of $\Delta = 0.6$, $\Delta = 0.7$, $\Delta = 0.8$, and $\Delta = 0.9$ respectively. These cases were performed comparing the MDI, Bouzidi quadratic, and Yu linear (MDI, BZ, and YU) boundary treatments. Again, like the 2D boundary treatment studies there were a total of 13 computations for these five cases. Table 4.11 lists the lattice Boltzmann parameters for all the test cases. For each case, the lattice spacing (Δx, Δy) was held constant at 0.1m while the length of the sides of the channel was varied by varying the resolution. The resolutions used to resolve the channel cross section were 3x3, 6x6, 9x9, 12x12, and 15x15. Therefore a total of 65 computations were performed (5 resolutions with 13 computations each). The flow in the lattice Boltzmann code was driven by a body-force that produces a maximum velocity of 0.1m/s. Again for simplicity, the average density was set at 1.0kg/m^3, while the relaxation constant was set at $\tau = 1$ for all cases.

Table 4.11 – Lattice Boltzmann Parameters for the 3D HCF Grids

Grid	Resolution	Lattice Spacing Δy (m)	Channel Dimension L (m)	Boundary Distances Δ	Boundary Treatments Tested
HCF1	3x3x21	0.1	0.3	0.5	SBB
HCF2	6x6x21		0.6	D1 = 0.6	MDI,BZ,YU
HCF3	9x9x21		0.9	D2 = 0.7	MDI,BZ,YU
HCF4	12x12x21		1.2	D3 = 0.8	MDI,BZ,YU
HCF5	15x15x21		1.5	D4 = 0.9	MDI,BZ,YU

The results were directly compared with the analytic solution of pressure-driven flow in a square channel (See Equations 4.1 – 4.4) in MATLAB. The RMS error was calculated using the same formulas previously used. Figures D.1 to D.4 illustrate the behavior of the RMS error versus $1/L^2$ for the D1, D2, D3, and D4 cases. The performance of the SBB boundary treatment for the $\Delta = 0.5$ case is plotted as a reference in all figures. Also note that all four figures are scaled identically on the RMS error axis. As previously seen in the 2D study, there is second order convergence of the error for all of the cases. Linear trend lines are added to the figures to help visualize this decay in error as the grid resolution is increased. Table D.1 summarizes the RMS error results for all of the cases computed. The results for 3D mirror the results for 2D in that the MDI and BZ boundary treatments do better than the YU boundary treatment. The results also point to the solution at $\Delta = 0.5$ being the most accurate with the lowest overall error. The results also show that as the boundary distance moves away from this midpoint, the RMS error tends to increase for all the interpolation boundary treatments.

Lid-Driven Cavity Flow

The flow in a 3D cavity driven by a moving lid was simulated using similar geometry and flow parameters used in the 2D boundary treatment study. The length of the side of the cavity was set at L = 24.18dm = 2.418m and the depth of the cavity in the z-direction was set to D = 4.98dm = 0.498m. As previously, the kinematic viscosity was set to υ = 0.01dm^2/s = 0.0001m^2/s, the lid velocity was set at U_1 = 0.413565dm/s = 0.0413565m/s, and the fluid density was set to 1kg/dm^3 = 1000kg/m^3. Because of memory limitations of the 3D lattice Boltzmann implementation, the resolutions used in the 3D grids are coarser than the finest 2D grids. Table 4.12 lists the parameters used for the three resolutions tested. The geometry of the cavity and the flow parameters are maintained by adjusting the lattice spacing, boundary distance, and relaxation parameter for each grid.

Table 4.12 – Lattice Boltzmann Parameters for the 3D LDCF Grids

Grid	Resolution	Lattice Spacing Δx (dm)	Boundary Distance Δ	Relaxation Parameter τ
LDCF1	81x81x17	0.30	0.300	0.600
LDCF2	101x101x21	0.24	0.375	0.625
LDCF3	121x121x25	0.20	0.450	0.650

For each resolution, solutions using the SBB, BZ, and YU boundary treatments were obtained. This resulted in 9 total computations (3 resolutions with 3 boundary treatments each) for this part of the study. For the SBB boundary treatments, the cavity walls were treated as though they were at a boundary distance Δ = 0.5, even though the physical geometry of the cavity differed. The other two interpolation boundary treatments could handle the varying boundary distances for each grid. The lattice

Boltzmann results from all of the LDCF computations are compared to a single FLUENT LDCF computation.

The analysis of the present study and further 3D studies comparing the lattice Botlzmann results with FLUENT results are similar is similar to the 2D analysis. In 3D, however, the entire domain of results cannot be compared directly with ease. Therefore results are compared only at certain z-planes that are chosen for the geometry. For these 2D planes, the x-velocities, y-velocities, and z-velocities can be compared directly as done previously in the 2D studies. For the 3D LDCF geometry, the z-planes Z = 1.2dm and Z = 2.4dm will be examined. The Z = 2.4dm plane is at the midpoint of the cavity with respect to the z direction and is a plane of symmetry. Therefore, z-velocities are expected to be negligible and are not analyzed. Figure 4.19 illustrates the Z = 1.2dm plane which is not at the midpoint of the cavity and should yield some results for the z-velocities. In the figure, the red arrow illustrates the direction of the lid velocity. Also note that the origin in absolute terms (0dm, 0dm, 0dm), is at the back lower left corner of the cavity.

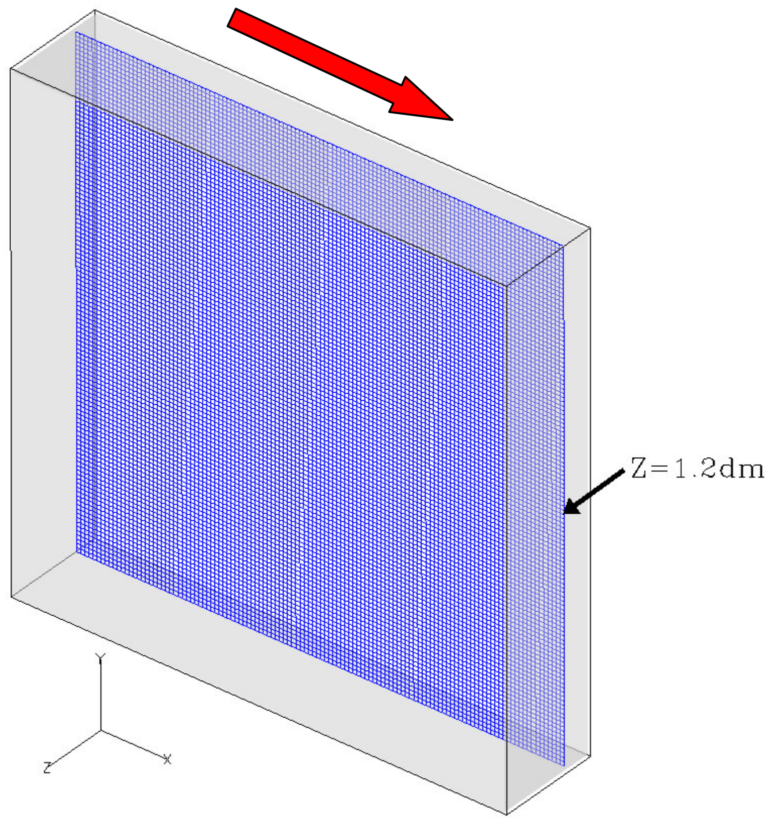


Figure 4.19 – Illustration of Z = 1.2 dm Plane for the LDCF Geometry

Streamline visualizations comparing the lattice Boltzmann and FLUENT results are presented in Figure D.5 and Figure D.6. These streamlines were calculated by seeding the Z = 2.4dm surface and remained relatively in the same spatial plane. Streamline visualizations for the Z = 1.2dm surface are not presented because of the three dimensional nature that the streamlines take on because of the z-velocities present at that surface. In the figures, the streamlines are colored by velocity magnitude. Figure D.5 is the FLUENT visualization while Figure D.6 is the lattice Boltzmann visualization using the LDCF3 grid. While the center vortex is comparable in both figures, a discrepancy is seen in the lower right vortex. The FLUENT calculation predicts a larger area of recirculation for that vortex while the lattice Boltzmann calculation predicts a much smaller vortex. This contrasts the 2D results where the lower corner vortices appear to be the same shape in both calculations.

Velocity line plots along the x and y centerline of the cavity on the Z = 1.2dm plane are presented in Figures D.7 to D.9. These plots are similar in nature to plots generated at Z = 2.4dm plane, so therefore those plots are not presented. Each of the three boundary treatments tested (SBB, BZ, and YU) is presented. The solution for the LDCF3 grid cases (dashed line) are compared against the FLUENT solution (solid line) directly. Results for the LDCF1 grid cases are not presented because they are very similar to the LDCF3 results. In the figures, the x and y position are normalized by the cavity length L and the velocities are normalized by the lid velocity U_1. While the x-velocity plots (shown in blue) essentially match the FLUENT solution, some differences in the performance of the boundary treatments can be seen on the y-velocity plots (shown in black). The SBB boundary treatment shows the least amount of deviation from the FLUENT solution for the y-velocity plots.

For the 3D studies, it is important to test the z-velocity results. Figures D.10 to D.12 are z-velocity plots along the x direction where Z = 1.2dm and Y = 21dm. For these plots, the x position is normalized by the cavity length L and the z-velocity is normalized by the lid velocity U_1. The results for the LDCF1 (open circles) and the LDCF3 (dashed line) are compared to the FLUENT results (solid line). In general the z-velocity from the lattice Boltzmann results tracks well with the FLUENT results. Also, it is clear that the refinement of the grid improves the accuracy. As in the previous velocity plots, the SBB treatment does a slightly better job than the other two interpolation boundary treatments.

To verify this quantitatively, a visual check of the absolute errors and an analysis of the velocity errors are performed.

Absolute error for the y-velocity in the LDCF3 grid cases is presented in Figures D.13 to D.15 for the SBB, BZ, and YU boundary treatments respectively. These figures support the conclusion made from the previous centerline plots results, in that the SBB boundary treatment does perform better than the two interpolation treatments. It is clear that the YU boundary treatment exhibits the most error in Figure D.15 and the SBB the least in Figure D.13. Further support of this absolute error data is given in Figures D.16 to D.20. These figures present the velocity RMS error plots for all the LDCF grid resolutions at both the Z = 1.2dm plane and the Z = 2.4dm plane. Note that the z-velocity for the 2.4dm plane is not presented because the z-velocities at that mid plane are essentially zero. While there is a decrease in RMS error as the grid is refined for all the boundary treatments, it is clear that the SBB boundary treatment performs much better in terms of x-velocity, y-velocity, and z-velocity.

Lastly, the percent error results are presented in Tables D.2 to D.6. Tables D.2 and D.3 list the data for all of the LDCF grid resolutions and all the boundary treatments at the Z = 2.4dm plane for the x-velocity and y-velocity respectively. Tables D.4 to D6 are similarly structured and list data for the x-velocity, y-velocity, and z-velocity for the Z = 1.2dm plane. These results show that the SBB treatment is again more accurate than the other two interpolation treatments with respect to the percentage of nodes with a percent error <1%, <5%, and <10%.

Flow in an Annulus

The boundary treatment studies done for the 3D flow in an annulus are structured in the same way the 2D studies were done. However, there are some differences that need to be noted. The geometry of the annulus is scaled down by a factor of 2 from the 2D geometry. This resulted in the inner radius measuring R_1 = 0.3045dm and the outer radius measuring R_2 = 0.876dm. The length of the annulus in the z-direction was set equal to 1.8225dm, which is slightly larger than the diameter of the annulus outer wall. The absolute coordinates of the center of the annulus were set to (0.9dm, 0.9dm, 0.9dm). As with the initial 2D boundary treatment studies, the kinematic viscosity and wall velocities were varied for an annulus without a construct for nine distinct cases. These are the same cases listed previously in Table 4.3. The density of the fluid was kept constant at 1kg/dm^3 = 1000kg/m^3. The resolutions used in the 3D computations are coarser than the 2D resolutions because of memory constraints. The grid resolutions used for the 3D studies are listed in Table 4.13. Thus, the results consisted of 81 computations (3 grid resolutions for 9 cases using 3 boundary treatments) performed for this part of the annulus boundary treatment study.

Table 4.13 – Lattice Boltzmann Parameters for the 3D Annulus Grids

Grid	Resolution	Lattice Spacing Δx (dm)
3D ANN1	41x41x41	0.0450
3D ANN2	61x61x61	0.0300
3D ANN3	81x81x81	0.0225

The stability of the lattice Boltzmann results for the 3D annulus are presented in Table 4.14. As previously seen in the 2D boundary treatment study results, there are several cases where computations were unstable. Since CAS9 parameters yielded an unstable solution in FLUENT in 2D, the parameters were not used in the 3D study. Comparing the results in Table 4.14 to the 2D results, the pattern of instability is roughly the same. The lattice Boltzmann computations tend to become more unstable for lower resolution, lower kinematic viscosity, and higher velocity.

Table 4.14 – Stability of Lattice Boltzmann 3D Annulus Results

Grid\Case	CAS1	CAS2	CAS3	CAS4	CAS5	CAS6	CAS7	CAS8	CAS9
ANN1	S	S	BZ	S	S	SBB,BZ	U	U	N/A
ANN2	S	S	BZ	S	S	BZ	S	S	N/A
ANN3	S	S	S	S	S	S	S	S	N/A

As in the 3D LDCF studies, results are compared only at certain z-planes that are chosen for the geometry. For the 3D annulus geometry, the z-planes Z = 0.9dm and Z = 0.27dm will be examined. For these 2D planes, the x-velocities, y-velocities, and z-velocities are compared directly. The Z = 0.9dm plane is at the midpoint of the annulus with respect to the z direction and is a plane of symmetry. For this plane z-velocities are negligible and are not analyzed. In contrast, for the Z = 0.27dm plane, higher z-velocities are seen. Figure 4.20 illustrates the Z = 0.27dm plane for the annulus geometry. The red arrow in the figure indicates the direction of the outer and inner wall rotation. Note that the back of the annulus is located at the absolute coordinate Z = 0dm.

Visualizations are presented for the annulus results. Figures D.21 to D.23 are typical absolute error plots for the 3D annulus geometry. The x-velocity error at the mid plane Z = 0.9dm is shown in Figure D.21 and is similar to what was seen in the 2D annulus. In contrast, the x-velocity error at the Z = 0.27dm plane, which is illustrated in

Figure D.22, is markedly different which may be due to its 3D nature. Figure D.23 is a plot of the z-velocity errors at the Z = 0.27dm plane.

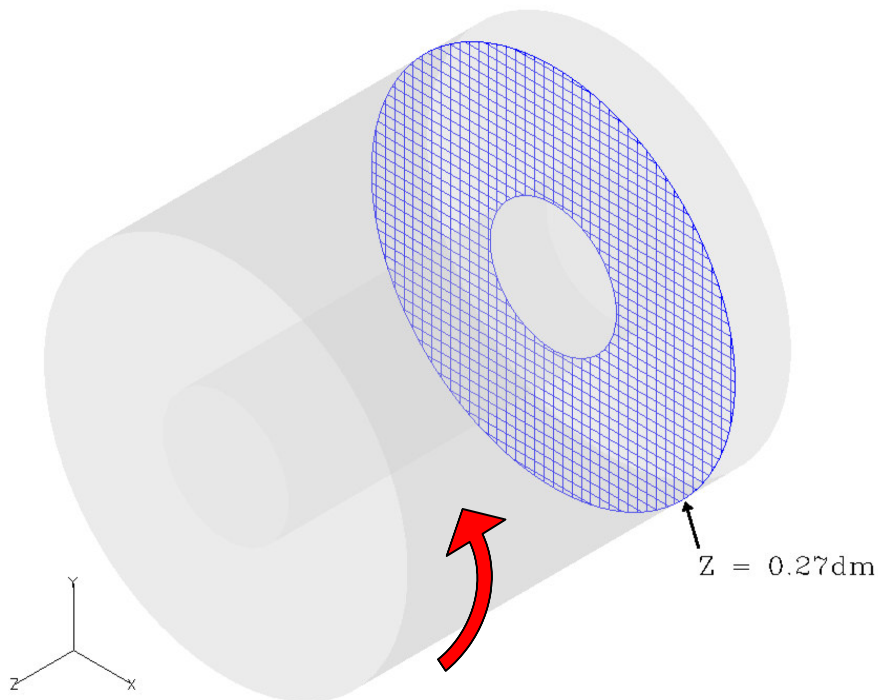


Figure 4.20 – Illustration of Z = 0.27 dm Plane for the 3D Annulus Geometry

To help demonstrate further the accuracy of the lattice Boltzmann 3D code, line plots of the y-velocity and z-velocity at the Z = 0.27dm plane are presented for the CAS5 flow parameters. The plots are done at the y midpoint of the annulus where Y = 0.9dm = 0.09m. Note that the wall velocity for this case is set to 0.1dm/s = 0.01m/s. The results for the three boundary treatments tested (SBB, BZ, and YU) are shown in Figures D.24 to D.29. The increase in resolution from the ANN1 grid to the ANN3 grid shows a qualitative improvement in accuracy for all cases.

The x-velocity and z-velocity RMS error results for the ANN3 grid are presented for the Z = 0.27dm plane in Figures D.30 to D.35. The x-velocity RMS error results for the SBB, BZ, and YU boundary treatments are presented in Figures D.30 to D.32, respectively. Likewise, the z-velocity RMS error results for those three boundary treatments are presented in Figures D.33 to D.35. The y-velocity results are not presented because they mirror the x-velocity results due to symmetry. In all the figures, the eight cases (CAS1 through CAS8) that varied flow parameters are presented. Also, note that the z-axes are scaled similarly for easy comparison. The x-velocity RMS error results for the Z = 0.9dm plane are similar to these results and are not presented.

The general trends for RMS error with increasing wall velocity as seen in the 2D implementation are also seen in the 3D annulus cases. An increase in velocity tends to produce more error given the same kinematic viscosity. However, in contrast to the 2D results, it is not clear that a decrease in kinematic viscosity increases error for the 3D cases. Another point of contrast with the 2D results is that when comparing boundary treatments, the SBB boundary treatment does almost as well as the BZ and YU boundary treatments. Thus, a better picture of boundary treatment performance is seen when looking at the entire velocity field rather than just the velocity line plots previously presented in Figures D.24 to D.29. Further insight into the accuracy can be had by examining the percent error results. The percent error results for the x-velocity and z-velocity are presented for the ANN grids using the CAS5 flow parameters at the Z = 0.27dm plane in Tables D.7 and D.8. In both tables, the SBB boundary treatment does slightly worse than the BZ and YU boundary treatments with respect to the <5% and <10% thresholds, but does notably worse at the <1% threshold.

The next set of computations done for the 3D annulus geometry placed a stationary construct in the flow. The constructs included a cube (CUBE), sphere (SPH), disc (DSC), and ellipsoid (EL). All constructs were placed at the three o'clock position, which is similar to the 2D studies. The cube was oriented with the main axes, while the main axis of the disc and ellipsoid were tilted at a 15 degree angle with respect to the x-axis. The disc construct and the ellipsoid construct were similar to their 2D counterparts (the rectangle and ellipse) in that their cross-sections were identically shaped. However, for the 3D cases, those 2D cross sections were then revolved around the short axis in order to form the 3D constructs. Thus the revolved rectangle created a disc, and the revolved ellipse created an ellipsoid. The diameter or characteristic length of the constructs was scaled to ½ of the value used in the 2D studies (since the 3D annulus is half the physical size of the 2D annulus) and was set to 0.204dm = 0.0204m. For this size construct, the number of lattice spacings across the construct ranges from 4 to 9 spacings for the lowest to highest resolution annulus grids. This only allows for a fairly coarse approximation of the construct boundary, however annulus geometry and memory limitations would not allow for a much finer resolution of the construct surface. For the computations performed with the constructs, the CAS5 flow parameters are used. This amounts to 36 total computations (4 constructs with 3 grid resolutions and 3 boundary treatments) performed for this part of the study. As done previously in the 3D studies, the analysis of the results will only be done for 2 specific Z planes. In this case, the midpoint Z = 0.9dm plane and the Z = 0.81dm plane will be examined. The z-velocity is small at the Z = 0.9dm plane since it is a plane of symmetry. Therefore, z-velocity will be examined at the Z = 0.81dm plane which primarily intersects the back edge of the constructs. Figure 4.21 shows the placement of the sphere construct within the annulus and where the Z = 0.81dm plane cuts across the construct. In the figure, the sphere construct is the darkened gray surface. The red area of the grid is where the construct boundary and the Z = 0.81dm plane intersect which is on the backside of the construct.

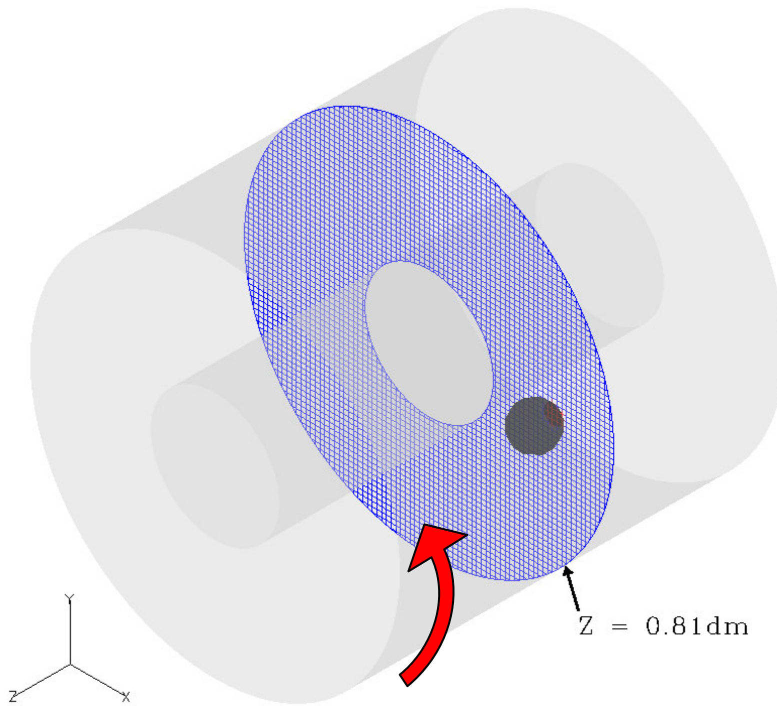


Figure 4.21 – Illustration of Z = 0.81 dm Plane for the 3D Annulus Geometry

The visualizations presented in Figures D.36 and D.37 are absolute error plots similar to those presented for the 2D annulus cases. Figure D.36 is a y-velocity error plot for the ANN3-CAS5 grid at Z = 0.9dm with the CUBE construct is placed in the flow. As previously seen in 2D, the highest velocity error occurs at the edges of the construct. Figure D.37 is a z-velocity error plot for the ANN3-CAS5 grid at Z = 0.81dm with the SPH construct placed in the flow. This is the same plane pictured in Figure 4.21. The highest z-velocity error is seen at the construct surface as well in Figure D.37. In both figures, the SBB boundary treatment results have been plotted.

Velocity RMS error results for the ANN-CAS5 construct cases are presented in Figures D.38 to D.40. These results are for the Z = 0.81dm plane. The x and y velocity results for the Z = 0.90dm plane are very similar to the Z = 0.81dm results and are not presented. The y-velocity RMS error versus construct type is presented in Figure D.38. Surprisingly, the SBB boundary treatment does better than the BZ and YU interpolation boundary treatments for all but one of the constructs. This contrasts previous results whereby the SBB boundary treatment was less accurate than the other boundary treatments. In Figure D.39, the y-velocity RMS error is plotted versus grid resolution for the SPH construct. The RMS error decreases as the grid resolution increases which is expected. The SBB boundary treatment again exhibits less y-velocity RMS error than the other two treatments. Lastly, the z-velocity RMS error versus resolution is presented for the SPH construct in Figure D.40. These results shows that the SBB boundary treatment does not perform as well for the z-velocity as it did with the y-velocity. The BZ and YU treatments perform better with regards to the z-velocity. However, since the order of magnitude of the y-velocity is greater than the z-velocity, overall it is likely that the SBB boundary treatment is more accurate. The percent error results for the corresponding cases are presented in Tables D.9 to D.11. The same conclusions can be drawn from the tables as from the previous figures. For the y-velocity in Tables D.9 and D.10, the SBB boundary treatment does better than the other boundary treatments, which can primarily be seen at the <1% percent error threshold. For the most part at the <5% and <10% percent error thresholds they perform equally. However in Table D.11, for the z-velocity the SBB boundary treatment does poorly at all threshold levels when compared to the BZ and YU boundary treatments.

Overall, the trends seen for the results of the 3D boundary treatment studies are similar with the results found for 2D. The velocity RMS error trends seen for the LDCF and annulus studies mirrored those trends seen in 2D. However, when comparing boundary treatments, the results also showed that the SBB boundary treatment performed better at these resolutions than the interpolation treatments, especially for the LDCF cases. Further examination of the SBB boundary treatment performance is necessary in the grid refinement and dynamics studies.

4.4.2 Grid Refinement Studies

The results of the 3D grid refinement studies are presented. The implementation of the grid refinement in 3D involved refining the grid in all three dimensions, but the analysis of the results was done only in the two dimensional Z planes mentioned earlier in the boundary treatment studies. The analysis will examine if the grid refinement actually increased accuracy of velocities around a construct for the 3D cases. The increase in accuracy should lead to better approximations of forces and shear stresses on the construct, which was seen in the 2D annulus grid refinement cases. The results of the numerical studies in this section are intended to only assess the accuracy of the grid refinement implementation in terms of velocities. The following section containing the results from the dynamics studies will assess the accuracy of the force and shear stresses when using grid refinement.

Grid Refinement for Lid-Driven Cavity Flow

Grid refinement was performed on the lid-driven cavity flow computations presented in the 3D boundary treatment studies. As done previously in 2D, the 3D geometry and flow parameters of the LDCF cases were preserved while additional grid refinement studies were done on the LDCF1 and LDCF3 grid resolutions. Multiple grid refinement computations were performed for the LDCF1 and LDCF3 cases where sub-grid resolutions were below the resolution of the finest main grid resolution. The same methods were used to implement the grid refinement in 3D as in 2D except that the refinement was extended in the Z directions. Therefore the sub-grid in 3D was a grid resembling "cube" rather than a simple 2D square as before. The 3D sub-grid also can be placed anywhere "on top" of the main grid. The grid refinement computations done for each main grid resolution is listed in Table 4.15.

Table 4.15 – Grid Refinement Computations for the 3D LDCF Main Grid Resolutions

Grid	Main Grid Resolution	Lattice Spacing Main Grid Δx (dm)	Refinement Factor	Lattice Spacing Sub-grid Δx (dm)
LDCF1	81x81x17	0.30	3	0.10
			6	0.05
LDCF3	121x121x25	0.20	2	0.10
			4	0.05

For the grid refinement studies presented in the table, the highest refinement of the LDCF1 or LDCF3 main grid results in a lattice spacing for the sub-grid of Δx = 0.05dm. This is equivalent to a main grid resolution of around 481x481x97. Thus for the 3D cases, four separate computations across the two main grid resolutions were performed to examine the effectiveness of refining the grid. The same three areas of refinement were chosen for the 3D lid-driven cavity flow case: the upper left (UL) corner of the cavity, the upper right (UR) corner of the cavity, and the lower middle (LM) of the cavity. The grid refinement done in these areas included both the Z = 1.2dm and Z = 2.4dm planes. This allowed for an analysis to be performed that can be easily compared to the 3D boundary treatment results presented in the previous section. For each of these areas of refinement, three boundary treatments (SBB, BZ, and YU) were tested to highlight the differences in accuracy. Thus, 36 total computations (3 areas of grid refinement with 4 sub-grids using 3 boundary treatments) were completed for this section.

Absolute error visualizations are presented in Figures D.41 and D.42 for the LDCF3-UL-RF4 case using the BZ boundary treatment. The figures show the absolute x-velocity error that occurs at the Z = 1.2dm plane in the UL corner of the cavity. Figure D.41 illustrates the reduction of error on the sub-grid when comparing it to the main grid error shown in Figure D.42 (Note that the z-axes are scaled identically). This reduction in error also occurs for the refinement cases involving the UR corner for the boundary treatments tested.

The RMS error for the x-velocity, y-velocity, and z-velocity results are graphed for various 3D LDCF cases in Figures D.43 through D.48. The graphs list the RMS error for the refined sub-grids (RF6 and RF3 in the case of the LDCF1 main grid and RF4 and RF2 in the case of the LDCF3 main grid), the overall RMS error for the entire main grid, and the RMS error of the main grid area covered by the sub-grid (SGA). In Figures D.43 and D.44, the x-velocity RMS error for the LDCF1-UL and LDCF3-UL cases at the Z = 2.4dm plane are presented respectively. For these cases involving the UL corner, the sub-grid RMS error for the finest sub-grid begins to approach that of the entire main grid.

This was also seen in the 2D results. The y-velocity RMS error for the LDCF1-UR and LDCF-UR cases are presented in Figures D.45 and D.46. These calculations were done for the results on the Z = 1.2dm plane. From these figures it is apparent that the UR corner contains the largest velocity errors for the entire grid as the error does not dip below the average error of the main grid. For the same Z plane, the z-velocity RMS errors are plotted for the LDCF1-UR and LDCF3-UR cases in Figures D.47 and D.48. These errors are much lower than the x-velocity and y-velocity errors due to the smaller magnitude of the z-velocity.

Some trends in these figures are apparent with regards to grid refinement and boundary treatments. As the sub-grid is refined (going from RF3 to RF6 for the LDCF1 cases and from RF2 to RF4 for the LDCF3 cases), there is a drop in the RMS velocity error for nearly all the velocity types and cases. This is similar to the 2D results previously presented. With regards to boundary treatment, there is an interesting result that is apparent. While the SBB boundary treatment seemed to do much better than the BZ and YU interpolation boundary treatments for the main grid RMS error results, the reverse happens when the sub-grid is implemented. The SBB boundary treatment actually performs worse than the two interpolation treatments tested. This contrasts the 2D results as well since it was unclear which boundary treatment performed better for the LDCF cases.

Tables D.12 to D.14 show the percent decrease in the sub-grid RMS error with respect to the SGA RMS error for the most refined cases presented in the previous figures. Table D.12 lists the percent decrease for x-velocity RMS error in the LDCF1-UL-RF6 and LDCF3-UL-RF4 cases. Note that these sub-grid cases have the same lattice spacing on the sub-grid although their main grid spacing is different (See Table 4.12). In Tables D.13 and D.14, the decrease in y-velocity and z-velocity RMS error for the LDCF1-UR-RF6 and LDCF3-UR-RF4 cases are listed. In the case of the x-velocity and y-velocity results, there is a substantial decrease in RMS error ranging from 20% to 80%. It is also clear that the SBB boundary treatment RMS error goes from being the best on the main grid (SGA error) to the worst at the sub-grid level. Thus, the decrease in RMS error for the SBB boundary treatment is lower than that of the BZ or YU interpolation treatments. The z-velocity results show only a slight decrease in RMS error or even an increase in one case. This may be due to the smaller magnitude of the z-velocity or the fact that a boundary in the z-direction is not contained within the sub-grid. Thus, little improvement is expected.

The percent error results for the 3D LDCF cases are presented in Tables D.15 to D.20. The tables are paired together to contrast the percent error results for the most refined sub-grids and the percent error results for the main grid solution in the area covered by the sub-grid (SGA results). Thus, Tables D.15 and D.16 show the sub-grid percent error results and the SGA percent error results respectively for the x-velocity in the LDCF-UL cases. Tables D.17 and D.18 are the percent error results for the y-velocity in the LDCF-UR cases, while tables D.19 and D.20 list the percent error results for the z-velocity in that same geometry. For the x-velocity and y-velocity results, more nodes exhibit less percent error for the sub-grid results versus the main grid SGA results. The SBB boundary treatment also does a reversal here, going from being the most accurate to the least. Surprisingly, the z-velocity results do not show an increase in accuracy but exactly the opposite. This may be due to the reasons already stated.

The results for the refinement of the lower middle (LM) part of the cavity, which contains no boundaries in the sub-grid, are the last items presented. As done in 2D, this area of the cavity was chosen in order to test the consistency of the sub-grid solution with the main grid solution. The results should verify that there is no significant degradation in accuracy when the sub-grid is implemented. Tables D.21 and D.22 are the RMS error results for the x-velocity and z-velocity respectively. Within the table, the sub-grid RMS error is compared with the main grid or SGA RMS error. In line with the 2D results, the x-velocity results show some degradation, but it is within a 3% threshold for the most part. The z-velocity results show a much larger percentage increase (due to a smaller z-velocity magnitude), but the order of magnitude of the RMS error increase is the same as the x-velocity RMS error increase.

Grid Refinement for Flow in an Annulus

Grid refinement for the 3D computations involving flow in annulus around stationary 3D constructs is performed. The computations done with the CAS5 flow parameters are used as a starting point for the 3D grid refinement studies on the ANN1, ANN2, and ANN3 main grids. Table 4.16 lists the refinement factors used for each of the main grid resolutions. All three main grid resolutions have a sub-grid refinement that results in a lattice spacing of $\Delta x = 0.0075$dm. This sub-grid lattice spacing is equivalent to having a total main grid resolution of 241x241x241 or 3 times finer than the ANN3 main grid. A total of seven grid refinements are done across the main grid resolutions.

Table 4.16 – Grid Refinement Computations for the 3D ANN Main Grid Resolutions

Grid	Main Grid Resolution	Lattice Spacing Main Grid Δx (dm)	Refinement Factor	Lattice Spacing Sub-Grid Δx (dm)
ANN1	41x41x41	0.0450	3	0.01500
			4	0.01125
			6	0.00750
ANN2	61x61x61	0.0300	2	0.01500
			4	0.00750
ANN3	81x81x81	0.0225	2	0.01125
			3	0.00750

The constructs used previously in the 3D boundary treatment studies included the cube (CUBE), sphere (SPH), disc (DSC), and ellipsoid (EL) constructs. These four constructs will again be examined using grid refinement. Three boundary treatments (SBB, BZ, and YU) were tested. Therefore, 84 computations (4 constructs with 7 sub-grids using 3 boundary treatments) were performed for this part of the study.

Figure D.49 is a visualization of the absolute y-velocity error for the ANN3-CUBE-RF3 refined grid case at the Z = 0.9dm plane. The dashed red line indicates the boundary of the cube construct for that plane. Figure D.50 shows the absolute y-velocity error for the ANN3-CUBE main grid case at the same Z = 0.9dm plane. Comparing the main grid to the sub-grid reveals only a slight improvement at first glance. However, more analysis is necessary to evaluate the performance of the grid refinement. The RMS error is a better indicator if y-velocity error is reduced overall.

Plots of the velocity RMS error are presented in Figures D.51 to D.56. Only a portion of the results for this study are presented here and are representative of majority of the other cases performed involving other construct types. These plots are similar to the previous RMS plots which illustrated the velocity RMS error for the sub-grids, the

RMS error on the entire main grid, and the SGA RMS error for all the boundary treatments tested. In Figures D.51 and D.52, the y-velocity RMS error results are presented for the ANN1-CUBE and ANN3-CUBE cases respectively, for the Z = 0.90dm plane. Figures D.53 and D.54 present the x-velocity RMS error results for the ANN1-SPH and ANN3-SPH cases respectively for the Z = 0.81dm plane. Lastly, the z-velocity RMS error results for those same SPH cases are displayed in Figures D.55 and D.56. Note that for each pair of figures, the z-axes are identically scaled for easy comparison.

The trends in velocity RMS error are similar to those seen in the 2D studies. There is a decrease in error going from the SGA RMS error values to the sub-grid RMS error values especially for the ANN1 cases. Within the refined grid results, there is a slight decrease with increasing refinement in most cases. This is only slightly noticeable in the figures. The SBB boundary treatment, which exhibits the least amount of error on the main grid annulus cases, performs well for the refined grids in relation to the other boundary treatments. The BZ and YU interpolation boundary treatments perform consistently better for CUBE construct cases, but in general they perform comparably to the SBB boundary treatment for the remaining cases.

A comparison of the SGA RMS error and the sub-grid RMS error is presented in Tables D.23 to D.25 for the most refined sub-grids. Note that all of the sub-grids presented have the same lattice spacing. The percent decrease in y-velocity RMS error for the ANN-CUBE cases is presented in Table D.23. The percent decrease in x-velocity and z-velocity RMS error for the ANN-SPH cases are presented in Tables D.24 and D.25 respectively. One observation from the tables is that the percent decrease in RMS error tends to decrease as the main grid resolution increases. This is expected since the main grid accuracy increases which consequently diminishes the benefits of the sub-grid. In addition, the SBB boundary treatment seems to perform comparably in terms of RMS error to the other treatments in the case of the SPH construct, but not the CUBE construct.

In Tables D.26 to D.31, the percent error data for the most refined grids in the ANN-CUBE and ANN-SPH cases are presented. These tables are paired in order to show the difference in the percent error data for the sub-grids versus the main grid that covers the sub-grid area (SGA results). For example, Table D.26 is the y-velocity percent error results for the ANN-CUBE sub-grids, while Table D.27 is the percent error results for the ANN-CUBE SGA cases. The tables show that in nearly all cases, the percentage of nodes with a percent error below the thresholds (<1%, <5%, or <10%) increases for the sub-grids compared to the main grid. The percent error results also confirm the RMS error results that show the SBB boundary treatment performing on par for the SPH cases. This somewhat contrasts the 2D results where the SBB boundary treatment was a consistent underperformer.

The results from the 3D grid refinement studies have supported the conclusions drawn from the results of the 2D studies. The implementation of the grid refinement again shows a reduction in velocity RMS error for the LDCF and annulus 3D cases as the refinement increases. While the SBB boundary treatment showed promise in the main-grid boundary treatment studies, in the grid refinement studies it performed at the same level as interpolation treatments for the grid refinement studies.

4.4.3 Dynamics Studies

The 3D dynamics studies presented here test the accuracy of the 3D lattice Boltzmann model and implementation with respect to forces, shear stresses, and particle dynamics. The structure of the studies is similar to the 2D dynamics studies performed. The evaluation of forces and shear stresses will be done using the annulus geometry with a stationary construct. This data will be compared to the data taken from a 3D FLUENT computation which matches the geometry and parameters of the annulus problem. The dynamics problems will include the sedimentation of a sphere and the motion of a neutrally buoyant particle in the annulus. The results of these studies involving a moving boundary will be compared to previously published experimental or computational data when available.

Force Evaluation on a Stationary Construct

As in the 2D studies, the force on the construct is evaluated using the Aidun method (Equation 2.50) for SBB boundary treatment computations and the modified momentum exchange method (Equation 3.1) for the interpolation boundary treatment computations. Previously, it was noted that for the 2D computations the Aidun method is not as accurate when used with the interpolation boundary treatments and the modified momentum exchange method is not as accurate when used with the SBB boundary treatment. This makes sense since the Aidun method is based on shifted bounce-back and the modified momentum exchange method is based on interpolation. The computations that are used to evaluate the force on the construct are the steady state solutions previously presented for the boundary treatment studies and the grid refinement studies. For the boundary treatment studies, 36 computations were performed on stationary constructs in an annulus under CAS5 flow parameters. For the grid refinement studies, 84 computations were performed under the same CAS5 flow parameters and annulus geometry. For this part of the dynamics studies, the forces and moments are evaluated for each computation are compared to the FLUENT results for force. The constructs under consideration are the cube (CUBE), sphere (SPH), disc (DSC), and ellipsoid (EL) constructs. The forces and moments under consideration are identical to the 2D forces considered, namely the x-direction force, y-direction force, and z-torque. Since the constructs are symmetrical and placed in the z midpoint of the annulus, the z-direction force, the x-torque, and y-torque are small and therefore neglected in the analysis.

The force and moment results for the 36 main grid computations are listed in Tables D.32 to D.35 for the four constructs tested. The tables list the FLUENT results for that particular construct, the results for each boundary treatment for all main grids, and the percent error of the lattice Boltzmann calculation if the FLUENT result is taken as the solution. Overall, there is improvement in the accuracy as the main grid resolution is refined from ANN1 to ANN3. Also, the y-direction force is consistently the most accurate across constructs, most likely because the y-velocity dominates the flow around the construct and the y-direction force has the largest magnitude. The z-torque is the least accurate across the constructs, especially for the EL construct where percent error is consistently over 100%. This may be due to the small magnitude of the z-torque for that particular construct. With regards to the boundary treatments, the SBB boundary treatment calculations are consistently more accurate than the BZ and YU boundary treatment calculations for all constructs. This contrasts the 2D results, where the SBB boundary treatment was outperformed by the interpolation treatments in most cases.

The sub-grid results for the force and moment on stationary constructs is presented in Tables D.36 to D.39. The sub-grids results presented correspond to the finest refinement factors for each main grid resolution. In the 2D studies, there was a clear improvement in the force and moment results generated by the sub-grid calculations in comparison to the main grid calculations. For the 3D results, this is not the case for the x-direction and y-direction force. However, there is clear improvement with the z-torque calculations as the percent error for these calculations dropped for all boundary treatments and across all constructs. The failure of the sub-grid results to improve on the main grid results for the x-direction and y-direction forces is unexpected. The two possible causes may be either the error introduced by the grid refinement itself or the error in the FLUENT results taken as the solution. A more refined FLUENT solution (which had already been refined once and contained 1M cells and 200K nodes) was attempted, but due to computer memory constraints was not obtained. The introduced error in the 3D grid refinement may be small (3% RMS error was induced for the LDCF-LM cases), but may be enough to influence the force results. However, this was not the case in the 2D grid refinement results where a very fine FLUENT mesh was used for the solution. The most consistent performer across the forces and moments for the boundary treatments is the SBB boundary treatment. This along with the main grid results suggests that for calculating forces and moments in the 3D lattice Boltzmann model, the SBB boundary treatment gives better results.

Shear Stress Evaluation for Flow in an Annulus

The accuracy of the velocities and velocity gradients for the 3D lattice Boltzmann model will determine the accuracy of the shear stresses (σ_{xy}, σ_{xz}, and σ_{yz}) and the shear rate (*S*) calculated from that model. As done previously in 2D, the previous studies involving the stationary construct in the annulus are examined. This includes the main grid and sub-grid data for all boundary treatments. The lattice Boltzmann data is again compared to FLUENT data for shear stresses in the fluid and the shear rate for the same 3D geometry and flow parameters. Line plots of the shear stresses and shear rates plotting the lattice Boltzmann data versus the FLUENT data are done where Y = 0.9dm and Z = 0.9dm or where Y = 0.9dm and Z = 0.81dm. These lines cut across the annulus between X = 1.215dm to X = 1.755dm and intersect the construct. Figure 4.22 illustrates the Y = 0.9dm, Z = 0.9dm line along which some of the line plots are made. The grid shown is a slice of the FLUENT 3D mesh used with the ellipsoid construct colored blue. The red arrow indicates the direction of the outer wall motion. The FLUENT grid is refined near the construct in order to resolve the flow.

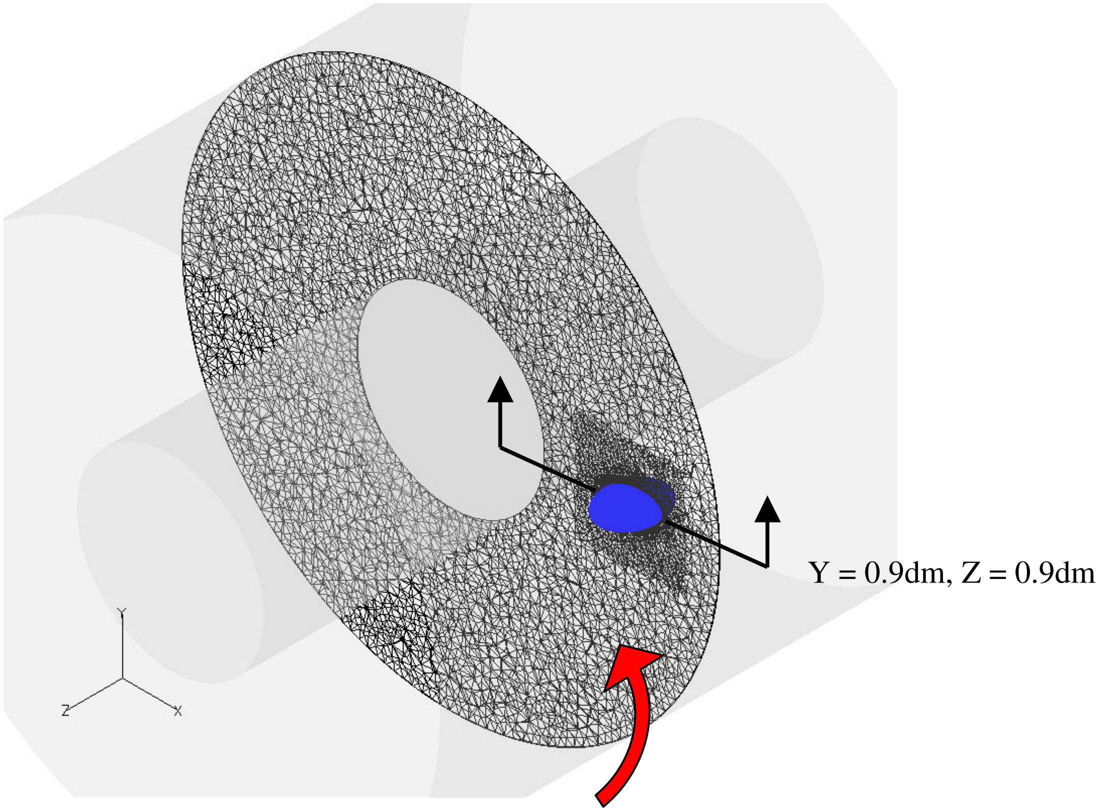


Figure 4.22 – Illustration of Line Plot at Y = 0.9dm, Z = 0.9dm

A qualitative comparison of x-y shear stress and the shear rate are presented in Figures D.57 to D.60. The FLUENT results for the EL construct are compared to the lattice Boltzmann results for the ANN3-EL-RF3 results using the BZ boundary treatment for the Z = 0.9dm plane. The ellipsoid surface is colored by either the x-y shear stress (units of dN/dm^2) or the shear rate (units of 1/s). Note that the lattice Boltzmann results show the sub-grid mesh. The lattice Boltzmann results in general compare well with the FLUENT results. However, from the figures it is clear that the FLUENT results show higher x-y shear stress and shear rate on the ellipsoid surface (color scales are matched). This may be due to the fact FLUENT calculates values exactly at the surface, while the lattice Boltzmann only gives values at lattice nodes inside the fluid. One other observation is that the FLUENT results create jagged contours compared to the smoother contours of the lattice Boltzmann computation. This may be a function of the triangular mesh that was used in FLUENT. To further examine the results quantitatively, line plots are presented.

A comparison between main grid and sub-grid shear stress and shear rate values for the ANN3-SPH case using the BZ boundary treatment is presented in Figures D.61 to D.64. These plots are along the Y = 0.9dm, Z = 0.81dm line since the results for the Z = 0.9dm line gives negligible values for the x-z and y-z shear stresses due to symmetry. The x-y, x-z, and y-z shear stress plots are presented in Figures D.61 to D.63 respectively, and the shear rate plot is shown in Figure D.64. Note that the construct boundaries exist around X = 1.45dm and X = 1.535dm. In the figures, the values from the ANN3-SPH main grid (open squares) are plotted with the ANN3-SPH-RF3 sub-grid values (open triangles) and both are compared to FLUENT values (solid green line). From the plots there is some improvement in the shear stress estimation, going from main grid to sub-grid values. This can especially be seen in Figure D.62 for the x-z shear stress. The shear rate plot shows that the maximum shear rate which occurs at the construct surface is higher than the maximum lattice Boltzmann sub-grid values. This may again be due to the fact that the FLUENT calculations are done at the surface while the lattice Boltzmann only calculates at the lattice nodes within the fluid. However, the error is most likely due to the lattice Boltzmann solution not tracking as well with the FLUENT solution. The 2D results showed very close results indicating that the velocities matched well. For the 3D results, the FLUENT and lattice Boltzmann velocities do not match as well and therefore more error is seen in the shear stress and shear rate calculations. An indication of this may be the jagged nature of the FLUENT results and may indicate refinement of that solution is necessary.

Additional line plots are presented in Figures D.65 to D.66. These plots examine the effects of the main grid solution on the sub-grid accuracy of the shear stresses and shear rates at the surface of the DSC construct using the BZ boundary treatment. These plots are again along the Y = 0.9dm, Z = 0.81dm line for the DSC surface at about X = 1.535dm. At the construct surface there are higher regions of velocity change and therefore is a much more interesting test of the lattice Boltzmann. The plots show the sub-grid solutions that have equal lattice spacing (Δx) for both the ANN1 (triangles) and ANN3 (lines) main grids. In both Figures D.65 and D.66 the main grid solution has little effect on the results. This contrasts somewhat the 2D results. However, it is clear that there is a marked difference in results between the sub-grid resolutions themselves. For example in Figure D.65, the finer sub-grid solutions (ANN1-RF6 and ANN3-RF3) track slightly better with the FLUENT solution than the coarser sub-grid solutions (ANN1-RF4 and ANN3-RF2).

The last set of line plots presented examines the differences in shear stresses and shear rates seen at the surface of the DSC construct due to different boundary treatments. The plots in Figures D.67 and D.68 are also along the Y = 0.9dm, Z = 0.81dm line for the DSC surface at around X = 1.445dm. The results are from the most refined sub-grid ANN3-RF3 case. The boundary treatments plotted include the SBB (open squares) boundary treatment, the BZ (open diamonds) boundary treatment, and the YU (+) boundary treatment. Surprisingly, the SBB boundary treatment does very poorly at the surface of the construct and gives an x-z shear stress and shear rate value well below the

FLUENT results. This was typical for the shear stress and shear rate plots done comparing the boundary treatments. The BZ and YU interpolation boundary treatments performed almost identically in most cases as well.

Lubrication Force between Two Approaching Spheres

The first dynamic case will measure the forces calculated by the lattice Boltzmann for two spheres approaching each other in a channel. The case is similar to the one performed for the 2D studies that used approaching cylinders. This case was also performed by Ding & Aidun (2003), and in that study the channel had a constant resolution of 64x32x32. The spheres had a constant radius of R = 4.25 lattice spacings where $\Delta x = \Delta y = 1$. The Reynolds number was held at 0.57 with the kinematic viscosity of the fluid and velocity of the spheres being varied for four cases. Again all of the parameters are expressed as non-dimensional. The non-dimensional force ($F/2R\upsilon\rho U$) between the spheres is calculated and plotted versus the non-dimensional gap size S. In this study, the same parameters are used and the cases are essentially identical. Table 4.17 shows the parameters used for this study and the 4 cases examined. Computations were performed for the SBB and BZ boundary treatments. Thus, for this part of the study 8 computations were performed (1 grid resolution, 2 boundary treatments, 4 cases).

Table 4.17 – Lattice Boltzmann Parameters for the 3D Lubrication Grids

Resolution	Lattice Spacing Δx	Construct Velocity U/2	Kinematic Viscosity υ
64x32x32	1	0.0025	1/12
		0.005	1/6
		0.01	1/3
		0.02	2/3

The analytic value for the squeeze film lubrication force is taken from previously published solutions. Equation 4.9 shows the calculated lubrication force $\mathbf{F_l}$ between two approaching spheres. This is essentially Equation 3.5 with the radius of the two spheres equal. Also, this is the same equation used by Ding & Aidun (2003). In Equation 4.9, the relevant parameters are the kinematic viscosity υ, fluid density ρ, relative approaching velocity $\mathbf{U}$, cylinder radius R, and gap size ε. Again, this lubrication force is used when it becomes larger than the calculated force between the two particles.

$$\mathbf{F_l} = -\frac{3}{2}\pi\upsilon\rho\mathbf{U}\frac{R^2}{\varepsilon} \tag{4.9}$$

Figures D.69 and D.70 show the non-dimensional force F = $\mathbf{F_l}/2R\upsilon\rho\mathbf{U}$ versus the non-dimensional gap size S (where S is gap size ε divided by the radius R) for computations using the SBB and BZ boundary treatment respectively for the various cases. The analytical solution is plotted on these figures as well. The results for the different boundary treatments were very similar with little difference. The calculated force does become smoother as the velocity increases. The roughness is most likely due to the rough grid approximation for the sphere affecting the force calculation at low velocities. These results from the lubrication cases again show that the current implementation allows for the correct force to be used when 3D spherical particles are in close contact.

Particle Sedimentation in a 3D Channel

The sedimentation of a sphere in a square vertical channel is modeled. Experimental results for the same problem were previously published by Miyamura et al. (1967). In the experiments, ball bearings with various diameters were dropped into a square channel containing a viscous fluid. Computational results for the problem also have been presented by Aidun et al. (1998). In those results, the dimensionless quantity $E = U_t/U_o$ was calculated using the terminal velocity of the sphere (U_t) and the unconfined terminal velocity (U_o) from the Stokes equation for drag on a sphere. Several calculations were done that varied grid resolution and the ratio of the diameter of the sphere to the channel side length (d/L).

In this study, several grids of varying resolutions are used to cover a similar range of d/L values used by Aidun et al. Table 4.18 lists the lattice Boltzmann parameters for the 3D sedimentation studies. The lattice spacing (Δx) and the construct diameter (d) is constant for all grids. Note that this combination yields approximately 10 lattice spacings across the diameter of the sphere. As done in the 2D studies, the density of the fluid was set to $1 kg/dm^3 = 1000 kg/m^3$ and the kinematic viscosity is set at $0.001 dm^2/s = 0.00001 m^2/s$. This in turn set the dynamic viscosity μ to $0.001 kg/dm \cdot s$. The length of the channel is about 20d, which means that the inlet and outlet of the channel are about 10d from the sphere. While a longer channel is more desirable, memory issues limited the length of the grids. The boundary condition used at the inlet and outlet is the free boundary condition or the Stokes boundary condition. The cavity boundary condition is not tested since the 2D results indicated very similar results to the Stokes boundary condition. Three boundary treatments (SBB, BZ, and YU) were tested for each grid resolution. Thus 12 computations (4 grid resolutions using 3 boundary treatments) were performed for the initial part of this study.

Table 4.18 – Lattice Boltzmann Parameters for the 3D VCF Grids

Grid	Resolution	Lattice Spacing Δx (dm)	Channel Side L (dm)	Construct Diameter d (dm)
VCF1	17x17x201	0.03	0.51	0.306
VCF2	25x25x201		0.75	
VCF3	37x37x201		1.11	
VCF4	51x51x201		1.53	

Grid refinement was applied to the sedimentation problem to examine its effects on the motion of the particle. The grid refinement results included an additional 12 computations for the single grid refinement factor applied. A refinement factor of 2 was applied to the main-grid resolutions which effectively doubled the node resolution around the particle. For both sets of computations, the particle position, particle forces, velocities, and terminal velocities are illustrated in the results presented.

The results from the sedimentation computations are displayed in Figures D.71 to D.76. Figures D.71 and D.72 plot the dimensionless constant E versus the d/L. The experimental best fit line published by Miyamura is also plotted. Figure D.71 plots the results from the first set of 24 main grid computations. Figure D.72 compares the results from the refine sub-grid using the BZ boundary treatment. Both plots show good agreement with the experimental results of Miyamura. The particle velocity and force plots for the VCF3 main grid and sub-grid are shown in Figures D.73 to D.76. Compared to the 2D results, these plots show less variation in velocity and force as the particle settles. Also similar to 2D, the terminal velocity of the particle increases when grid refinement is applied, as seen in Figure D.75. For the BZ boundary treatment, this brings the terminal velocity more in line with experimental results as seen in Figure D.72. In contrast, the results here show little variation across boundary treatments. Both the velocity and force plots are comparable for all boundary treatments.

Motion of a Neutrally-Buoyant Particle in a 3D Annulus

The motion of a neutrally buoyant sphere is examined for the 3D annulus geometry. The 3D studies done previously in the boundary treatment and grid refinement studies are used as a starting point for the calculations done here. The 3D ANN3 grid with a resolution of 81x81x81 is used in these calculations. The boundary treatments used are the SBB and BZ boundary treatments. The flow parameters are varied from the previous 3D annulus studies in order to produce rigid body rotation. These parameters are the inner and outer wall velocities. The outer wall linear velocity is set at 0.0185dm/s and the inner wall linear velocity is set at 0.0069dm/s for the 3D case. The kinematic viscosity, like in the 2D case, is set to 0.25dm^2/s. The steady solution is first computed and then the particle is released and the motion observed. Figure D.77 plots the particle path for the SBB and BZ boundary treatments after the particle is released. From the plot, the particle is seen returning to its original starting place. This is inline with the previous experimental results from Botchewey et al. (2000) and Wereley and Lueptow (1999), which were already mentioned.

Overall for the 3D dynamics studies, the lattice Boltzmann performs fairly well with respect to calculating forces and shear stresses on a construct. While the force and shear stress accuracy in 3D is not as dead on as the 2D results, this may be attributed to a couple factors including the accuracy of the refined FLUENT grid results and the coarse lattice Boltzmann grids used due to memory restrictions. However, for the studies involving dynamic motion it performs closely to the experimental results for the particle sedimentation studies. In addition, the result for the studies involving the neutrally buoyant particle in the annulus also were as expected. Based on these studies, the 3D lattice Boltzmann model implemented should give reasonable results for bioreactor simulations.

4.4.4 Simulations

Simulations of the RWV bioreactor are performed in 3D using the annulus geometry and the disc construct. In contrast to the 2D simulations, the 3D simulations should yield flow visualizations that are more in line with the actual flow seen inside the RWV bioreactor. Again the physical parameters of the RWV bioreactor model used by Neitzel et al. (1998) are followed. Like the 2D studies, the viscosity of the fluid could not be reproduced exactly. Table 4.19 shows a comparison of the parameters used for the 3D simulation and parameters in the study by Neitzel et al.

Table 4.19 – Lattice Boltzmann Parameters for the 3D Simulation

Parameter	LB Simulation	Neitzel et al. (1998)
Inner Wall Radius	0.1015dm	0.1dm
Outer Wall Radius	0.291dm	0.29dm
Inner Wall Velocity	13-14RPM	Varied (E) 13RPM (C)
Outer Wall Velocity	13-37RPM	Varied (E) 37RPM (C)
Construct Radius	0.034dm	0.02875-0.03285dm
Fluid Kinematic Viscosity	0.0005dm^2/s	0.0001dm^2/s = 1cS
Fluid Density	0.816kg/dm^3	0.816g/cm^3
Construct Density	0.830kg/dm^3	0.830g/cm^3

E – Experiments C – Computations

The lattice used in the 3D simulation is similar to the 3D ANN3 lattice used in the previous studies. In order to match the physical parameters listed in Table 4.19, it was scaled down by a factor of three. The lattice spacing was set to $\Delta x = 0.0075$dm for the main grid resolution of 81x81x81 lattice nodes. A sub-grid centered on the construct with a refinement factor of two was used. The relaxation constant corresponding to the desired kinematic viscosity was $\tau = 0.7$. The 3D disc construct was initially placed at the 3 o'clock position in approximately the same location as seen in the results of Neitzel et al. (1998). In generating simulation results, a steady state solution for the flow around the construct was first established. The construct was then released and its motion was observed. As with the 2D simulations, the intention was to match the motion of the construct in the simulation to the motion seen experimentally. This was done by either altering the net body force applied to the construct in the lattice Boltzmann simulations or by varying the wall angular velocities as done in the experiments by Brown (1998).

The results of the 3D simulations are presented for three cases. The first is the case where the motion of the construct is characterized by staying stationary in the flow. The second case is the case where the construct orbits around the annulus and the last is the case where the construct orbits about a fixed point. Figure D.78 is a velocity vector visualization of the flow around the rectangular construct before being released and is held at a stationary point. The vectors are colored by velocity magnitude and the scale is in dm/s. In contrast to the 2D results, the wake structure seen here is similar to the wake structure in the experiments and previous calculations by Neitzel et al. (1998). Figures D.79 to D.82 are the particle paths plotted for three cases mentioned. In Figure D.79 the path of the stationary construct takes it from the original release point to a point slightly lower in the annulus. The construct then remains stationary. Note that the net body force applied here is approximately equivalent to 1.7 times the force of gravity. This is most likely due to the difference in kinematic viscosity between the experiments and the simulation. In Figure D.80, the construct is seen orbiting around the annulus. The net body force applied in this case is much less than the first case. The most interesting particle path plot is shown in Figures D.81 and D.82. In this case, the wall angular velocities were altered in a similar fashion to the experiments. The inner and outer wall velocities were both set to 13RPM for the first case and 14RPM in the second case. The results show that the construct orbits in a circular motion around a point. This matches the behavior observed by earlier RWV bioreactor researchers and reproduced experimentally by Brown (1998). In Figure D.81 where the 13RPM is the wall angular velocity, the orbit is a tighter oval than compared to the particle path shown in Figure D.82 where the wall angular velocity is 14RPM. This expanding orbit was also observed by Brown in experiments where 13RPM and 15RPM cases were examined. Thus, the lattice Boltzmann simulations can match qualitatively the previously observed construct behavior.

Figures D.83 is a contour plot of the x-y shear stress for the third case mentioned. The maximum positive and negative shear stress values in the figure are close to 0.0278dN/dm^2. This is equivalent to a value of 2.78 dynes/cm^2, which is similar to the 2D simulation shear stress values obtained. Another way to estimate the average shear stress is to divide the drag force by the construct area. For this case, the drag force is 0.00025401dN and the construct surface area is 0.014529dm^2. These values yield an average shear stress of 0.017483dN/dm^2, which is equivalent to 1.75 dynes/cm^2. Again, this is within an expected range of shear stress values seen in previous results.

The results of the 3D simulations show that the lattice Boltzmann does reproduce both the structure of the flow and the behavior of the moving construct in the RWV bioreactor. Specifically, the wake structure seen in 3D is similar to that which is seen in previous experiments and computations. Also, the orbiting behavior of the construct in 3D matches the behavior seen previously. Remarkably, the circular orbiting behavior around a fixed point is reproduced by the lattice Boltzmann. The implementation also gives reasonable shear stress values near the construct, which is crucial in the growth process of the articular cartilage constructs. Thus, the lattice Boltzmann implementation presented here can be used to further examine the RWV bioreactor problem or other similar problems that involve moving particles.

4.5 Summary

This chapter details the numerical studies used to test the lattice Boltzmann implementation, the methods used in the evaluation of the results, and the results themselves. The numerical studies focused on testing the boundary treatment, grid refinement, and dynamics components of the lattice Boltzmann model. The evaluation of the results mostly involved comparing the velocity, force, and shear stress data generated by the lattice Boltzmann with results from identical cases done in FLUENT. In other instances, results were compared to previously published analytical, experimental, or computational results. Both the 2D and 3D results provided insight into the accuracy of the boundary treatments tested, the implementation of a moving sub-grid, and the motion of particles placed in the flow. Further discussion of the results follows in Chapter 5.

Chapter 5

Analysis and Discussion

The results presented in the previous chapter reflect the process of verifying the accuracy of the lattice Boltzmann code and the validation of the code in simulating the fluid dynamics in an RWV bioreactor. This chapter will first highlight the results that verify the accuracy of the code with respect to the methods used. The second part of this chapter details the validation of the code through simulations. The goal of this study is to produce an accurate and efficient lattice Boltzmann code. In the process of doing so, the lattice Boltzmann code has evolved due to the results obtained in these numerical studies. Accuracy and efficiency were taken into consideration in the use of certain lattice Boltzmann methods. For example, careful consideration was taken in deciding which boundary treatments are used in certain situations and how to evaluate the force on a surface. The discussion in this chapter highlights the key outcomes and results that verify and validate the lattice Boltzmann implementation.

5.1 Accuracy of the Lattice Boltzmann Implementation

The numerical studies presented in the previous chapter were designed to test the accuracy of the basic components of the lattice Boltzmann implementation. In testing the boundary treatments, grid refinement, and dynamics of the code, a good characterization of the methods is achieved. In addition, the best possible combination of methods can then be applied to the 2D and 3D simulations.

5.1.1 Boundary Treatments

A variety of boundary treatments have been proposed in the lattice Boltzmann literature. The shifted bounce-back boundary treatment is the most commonly used boundary treatment for it simple implementation. However, this treatment can only approximate boundaries that are not placed halfway between lattice nodes. Interpolation boundary treatments, such as those proposed by Yu et al. (2003) and Bouzidi et al. (2001), utilize linear or quadratic interpolation to account for the exact location of boundaries. This study has proposed a multi-dimensional interpolation boundary treatment that integrates more information to approximate complex boundaries. While these interpolation boundary methods account for the more complex boundaries, a comprehensive test comparing their performance with the shifted bounce-back boundary treatment has not been undertaken.

The numerical studies involving the boundary treatments, grid refinement, and dynamics have all tested the use of these boundary treatments. The MDI treatment was not tested for the majority of 3D studies however, due to memory and time constraints. Table 5.1 summarizes the performance of the boundary treatments with respect to the following criteria: practicality, stability, robustness, velocity accuracy, force accuracy, shear stress accuracy, and dynamics. The boundary treatments are ranked 1 to 4 for each criterion, and ties are considered where treatments performed comparably. While these rankings are subjective in nature, they are based on the quantitative results from the numerical studies. For most of the quantitative comparisons, the lattice Boltzmann results were compared directly to results done in the commercial CFD package FLUENT. Other results were compared with analytical solutions or experimental results.

Table 5.1 – Comparison of Boundary Treatments

	Practicality	Stability	Robustness	Velocity	Force*	Shear Stress	Dynamics
SBB	● ● ● ●	● ● ● ½	● ● ●	● ● ½	●	●	● ●
MDI*	●	● ½	● ● ● ●	● ● ½	● ● ● ½	● ● ●	N/A
BZ	● ● ½	● ½	● ½	● ● ½	● ● ● ½	● ● ●	● ●
YU	● ● ½	● ● ● ½	● ½	● ● ½	● ●	● ● ●	● ●

*Rankings based primarily on 2D performance

The practicality criterion is based on how easy a boundary treatment is to implement with regards to the relative speed it takes to execute and the memory requirements necessary for storage. The SBB treatment clearly was the easiest to implement while the MDI treatment was the most difficult. The MDI treatment required significantly more time to execute due to the interpolation it performed and required more memory in order to implement. Thus, the MDI treatment was not tested fully for the 3D lattice Boltzmann implementation. While the BZ and YU treatments performed similarly to the SBB treatment with respect to time, they did require tracking and storing the exact boundary position.

The stability rankings are based on the results from the studies involving flow in an annulus. Tables 4.5 and 4.14 illustrate the stability of the boundary treatments for 2D and 3D annulus cases that varied the wall velocity and the kinematic viscosity. In general, as the Reynolds number increases, either through an increase in velocity or a decrease in viscosity, the lattice Boltzmann method becomes unstable. The lattice

Boltzmann method also is more unstable for lower resolution grids (which require lower relaxation constants for a given viscosity). However, the stability of the method also depends on the boundary treatment given the same Reynolds number and grid resolution. From the tables, it is clear that the SBB and YU boundary treatments were the most stable, while the BZ boundary treatment was the least stable. The BZ boundary treatment was also unstable for the case of the stationary square construct in the annulus. The MDI treatment performed slightly better than the BZ boundary treatment for the 2D studies. However, the higher order interpolation appeared to cause more instability in the code. Thus, since both the BZ and MDI treatments are based on interpolation that was higher order than linear interpolation, they are expected to perform similarly in 3D as well.

The robustness of the boundary treatments was based on the velocity accuracy of the lattice Boltzmann as the Reynolds number is increased. The specific cases used in the evaluation are the 2D annulus cases using the CAS5, CAS6, and CAS8 flow parameters. The CAS5 flow parameters (wall velocity set to 0.01m/s and kinematic viscosity set to 0.00001m^2/s) yielded base results for the percent of nodes with a percent error less than 1%. The CAS6 flow parameters decreased the viscosity to 0.000005m^2/s while keeping the wall velocity at 0.01m/s. In contrast, the CAS8 parameters increased the wall velocity to 0.02m/s while keeping the viscosity to 0.00001dm^2/s. Table 5.2 summarizes the x-velocity percent error results for these cases using the ANN3 grid. The table clearly shows that the MDI treatment holds up well at the <1% threshold of percent error when comparing results from CAS5 to CAS6 or CAS8. The SBB boundary treatment actually improves, while the BZ and YU treatments do much worse when the Reynolds number increases.

Table 5.2 – X-Velocity Percent Error Results for the ANN3 Cases

Grid	Boundary Treatment	%nodes with PE < 1% CAS5	%nodes with PE < 1% CAS6	%nodes with PE < 1% CAS8
ANN3 121x121	SBB	89.81	91.57	91.99
	MDI	99.88	99.19	99.30
	BZ	98.43	41.32	59.10
	YU	99.07	47.30	66.53

The robustness can further be evaluated by examining the x-velocity RMS error for the same cases. Table 5.3 summarizes the x-velocity RMS error for the ANN3 cases. The MDI treatment maintains the RMS error as the Reynolds number is increased through a velocity increase (CAS8). It also does comparatively well when the viscosity is decreased (CAS6). The SBB treatment behaves in the exact opposite manner for these cases. The BZ and YU treatments both faired poorly, especially for the case where the velocity was increased. The interesting result from these studies is that for the same Reynolds number (CAS6 and CAS8), the lattice Boltzmann does not necessarily perform with the same accuracy.

Table 5.3 – X-Velocity RMS Error Results for the ANN3 Cases

Grid	Boundary Treatment	RMS Error (m/s) X-Velocity CAS5	RMS Error (m/s) X-Velocity CAS6	RMS Error (m/s) X-Velocity CAS8
ANN3 121x121	SBB	0.000037	0.000038	0.000070
	MDI	0.000011	0.000041	0.000014
	BZ	0.000026	0.000077	0.000115
	YU	0.000023	0.000068	0.000103

The rankings for the velocity accuracy of the boundary treatments were based on the velocity RMS error and velocity percent error results presented for the 2D and 3D annulus cases. The lid-driven cavity flow results were not used for several reasons. First, the 2D lid-driven cavity flow cases did not reveal a consistently better boundary treatment. Various boundary treatments performed best for different grid resolutions. Second, the 3D lid-driven cavity flow cases did show that the SBB treatment performed better overall, but the grid refinement studies for that same geometry showed that the BZ and YU treatments performed much better at finer resolutions. In part, these varied results are likely due to the velocity singularities that exist at the top corners of the cavity, the straight (non-complex) boundaries of the cavity, and the coarse nature of the 3D main grids. In such situations, the SBB treatment tends to do as well, or even better than other boundary treatments. But since this study is interested in evaluating boundary treatments for more interesting boundaries, such as curved boundaries, the annulus results were primarily used.

Consistently for the 2D annulus results, the interpolation treatments have resulted in lower velocity RMS error and lower percent error for grid nodes. For those same cases, the SBB treatment did not perform as well as the interpolation treatments. This not only occurred for the main grid calculations, but also the grid refinement calculations. Table 5.4 is a summary of velocity RMS error results for the 2D annulus cases with and without a construct. The results shown in the table are representative results for annulus main grid cases, construct cases, and refined grid cases. This is a small sampling of the results that were repeatedly seen for the numerous 2D computations performed. Table 5.5 is a summary of the percent error results for the same cases shown in Table 5.4. The percent error results for the <1% threshold mirror the RMS error results.

Table 5.4 – Velocity RMS Error Results for the 2D Annulus Cases

Grid	Boundary Treatment	RMS Error (m/s) X-Velocity CAS5	RMS Error (m/s) Y-Velocity CAS5-EL	RMS Error (m/s) Y-Velocity CAS5-CIR-RF4
ANN3 121x121	SBB	0.000037	0.000088	0.000153
	MDI	0.000011	0.000032	0.000061
	BZ	0.000026	0.000038	0.000045
	YU	0.000023	0.000041	0.000069

Table 5.5 – Percent Error Results for the 2D Annulus Cases

Grid	Boundary Treatment	%nodes with PE < 1% X-Velocity CAS5	%nodes with PE < 1% Y-Velocity CAS5-EL	%nodes with PE < 1% Y-Velocity CAS5-CIR-RF4
ANN3 121x121	SBB	89.81	44.07	7.52
	MDI	99.88	69.61	22.64
	BZ	98.43	60.59	41.57
	YU	99.07	60.81	19.88

The 3D annulus results reveal a slightly different story for the boundary treatments. Again, the MDI treatment was not fully tested for the 3D boundary treatment studies, but its velocity performance is assumed to be similar to the other interpolation treatments. Overall, the SBB treatment in 3D performed better on the main grid cases compared with its 2D results. The velocity RMS error results for most cases for the SBB treatment were equivalent to or lower than the interpolation treatments. For the most part concerning the percent error results, the SBB treatment performed well at the <1% threshold, but not as well at the <5% and <10% thresholds. The cases where the SBB treatment underperformed involved the CUBE construct or results calculated at the Z = 0.9dm plane for constructs. Tables 5.6 and 5.7 are representative of the 3D annulus results. The results for construct cases were done for the Z = 0.81dm plane, while the main grid results are for the Z = 0.27dm plane. In most cases, the SBB boundary treatment does well. However, as illustrated in the table, it may not always perform as the best boundary treatment for a given case. The interpolation treatments often performed equally well when compared to each other.

Table 5.6 – Velocity RMS Error Results for the 3D Annulus Cases

Grid	Boundary Treatment	RMS Error (m/s) X-Velocity CAS5	RMS Error (m/s) Y-Velocity CAS5-EL	RMS Error (m/s) Y-Velocity CAS5-DSC-RF3
3D ANN3 81x81x81	SBB	0.000082	0.000111	0.000134
	BZ	0.000080	0.000133	0.000135
	YU	0.000079	0.000125	0.000135

Table 5.7 – Percent Error Results for the 3D Annulus Cases

Grid	Boundary Treatment	%nodes with PE < 1% X-Velocity CAS5	%nodes with PE < 1% Y-Velocity CAS5-EL	%nodes with PE < 1% Y-Velocity CAS5-DSC-RF3
3D ANN3 81x81x81	SBB	20.47	42.47	26.03
	BZ	42.11	25.74	21.58
	YU	39.86	27.40	23.79

The final rankings for the velocity accuracy reflect the even performance of the boundary treatments. For 2D studies, the SBB treatment is the least accurate according to the results. However, for 3D studies the SBB treatment holds its own with the interpolation treatments in terms of velocity RMS and percent error results. Thus, the lattice Boltzmann model (either 2D or 3D) can influence the choice of an appropriate boundary treatment.

The force evaluation accuracy rankings are based on the 2D annulus construct studies force results. While the main grid force results were of good quality, the sub-grid results for the construct cases proved to be the most accurate. The percent error for the forces and torques evaluated (x-direction force, y-direction force, and z-direction torque) were evaluated across all the constructs (square, circle, rectangle, and ellipse). An average percent error was then taken to give a performance average for each boundary treatment. Table 5.8 lists those average percent errors for the most refined sub-grids done for each main grid.

Table 5.8 – Average Force Percent Error Results for the 2D Annulus Cases

Grid	Boundary Treatment	X-Force Average Error (%)	Y-Force Average Error (%)	Z-Torque Average Error (%)
ANN1 RF12	SBB	29.23%	13.50%	21.77%
	MDI	8.07%	8.95%	22.46%
	BZ	9.01%	9.37%	32.85%
	YU	20.33%	8.57%	14.36%
ANN2 RF6	SBB	36.06%	5.61%	12.62%
	MDI	25.20%	3.21%	8.61%
	BZ	27.74%	3.63%	7.11%
	YU	27.85%	3.62%	13.54%
ANN3 RF4	SBB	35.79%	5.78%	9.16%
	MDI	13.18%	3.67%	5.07%
	BZ	15.63%	3.85%	4.02%
	YU	14.45%	3.86%	8.24%

The results in Table 5.8 may be skewed higher by the percent error results for the more difficult construct boundaries to approximate (See Tables C.33 to C.36 for the full results); however the average percent error is a good overall indicator of performance. The SBB treatment does the worst for the 2D studies, while the MDI and BZ treatments perform better all around. This is also the case when scanning over the full 2D results. Note that the results for the 3D annulus construct studies were not used since a more refined 3D FLUENT solution may still be needed to obtain better force results. However, the 3D force results show (See Tables D.36 to D.39), much like the velocity accuracy results, that the SBB treatment does better with the 3D lattice Boltzmann model than the 2D model.

The shear stress accuracy was evaluated by plotting the calculated shear rate and shear stresses along a line that intersects a stationary construct within an annulus. The boundary treatment rankings are based on the results of line plots for both 2D and 3D cases. The primary constructs examined were the ellipse construct in 2D and the disc construct in 3D. These constructs were chosen because the velocity gradients were high at the surface of the constructs. Figures 5.1 to 5.4 are the results of the shear stress and shear rate plots across the 2D ellipse and 3D disc constructs. The shear stress and shear rates were calculated from the sub-grid solutions for the 2D ANN3 main grid and the 3D ANN3 main grid. Note that the construct boundary begins around X = 2.8dm for the 2D case and X = 1.45dm for the 3D case.

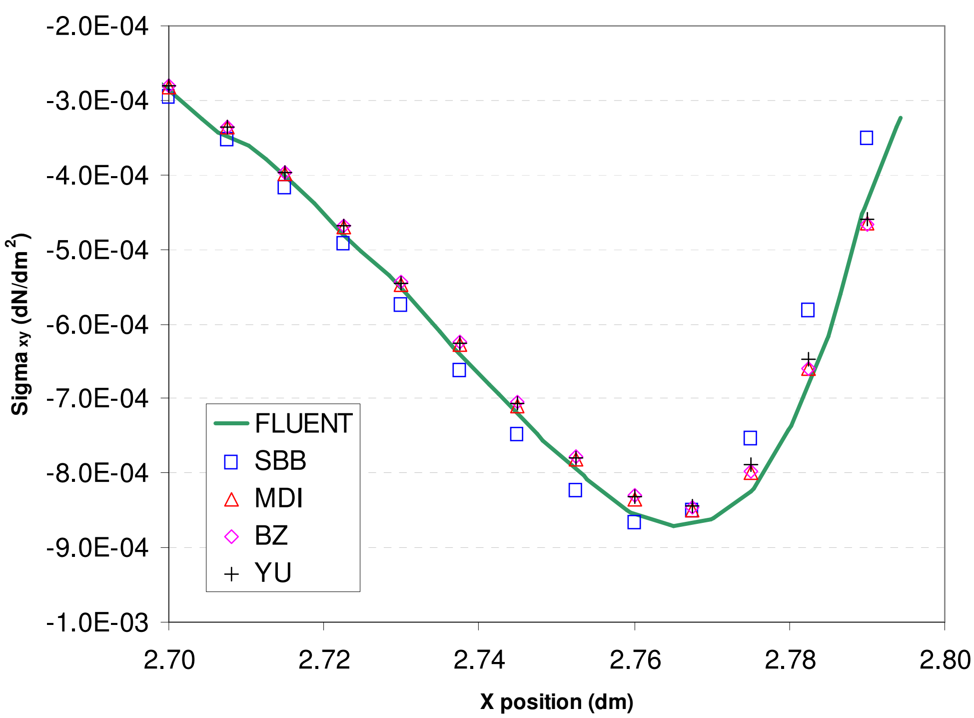


Figure 5.1 – Shear Stress Plot for the 2D Ellipse comparing Boundary Treatments

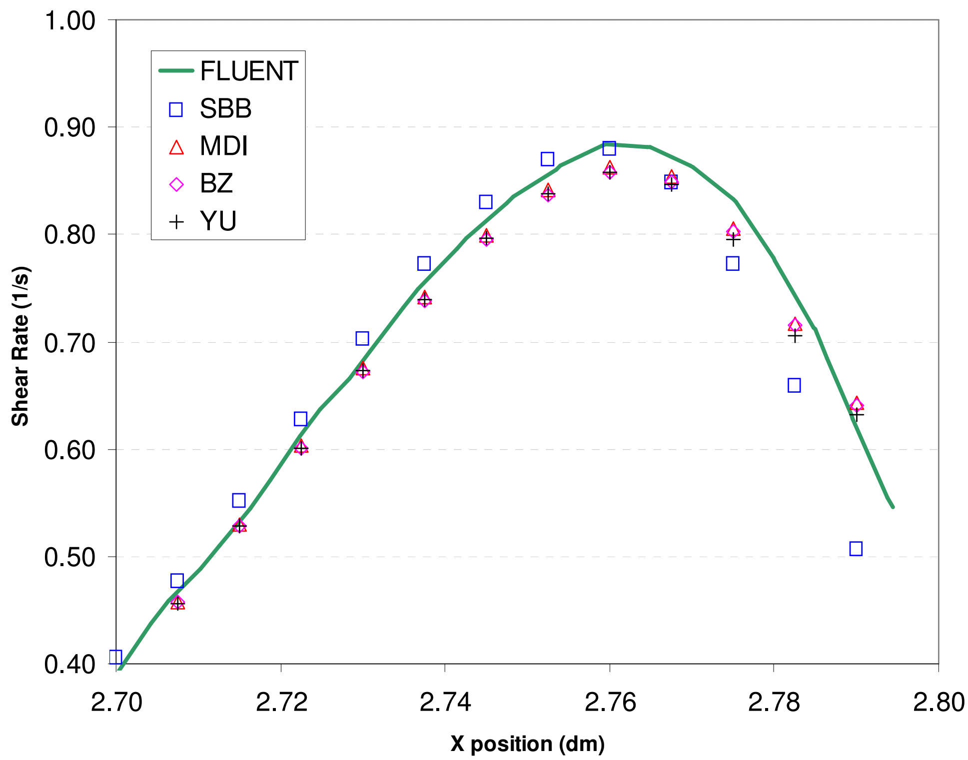


Figure 5.2 – Shear Rate Plot for the 2D Ellipse comparing Boundary Treatments

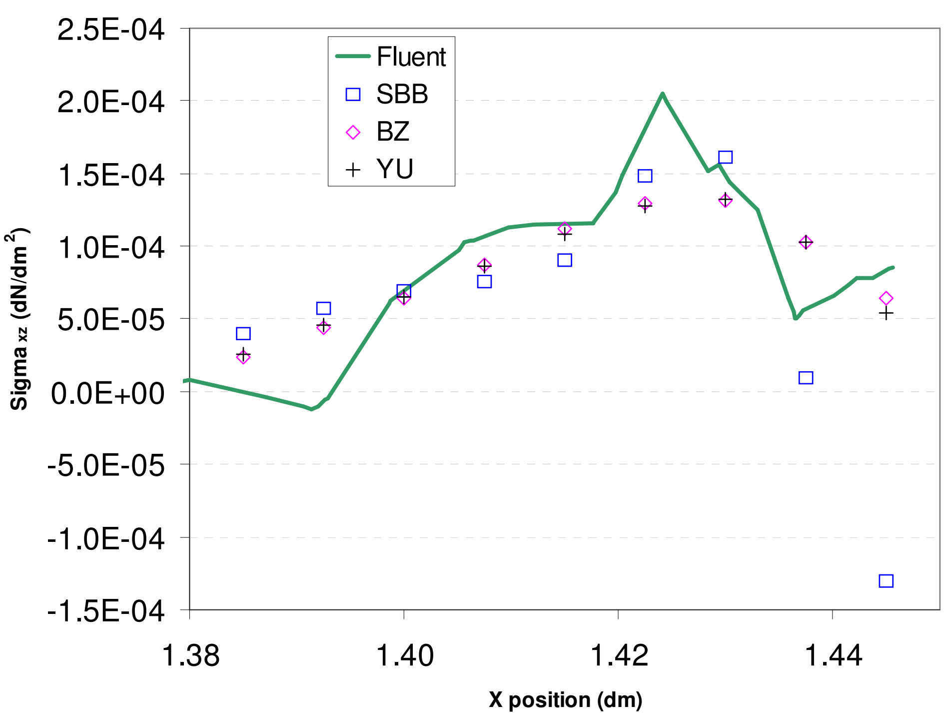


Figure 5.3 – XZ Shear Stress Plot for the 3D Disc comparing Boundary Treatments

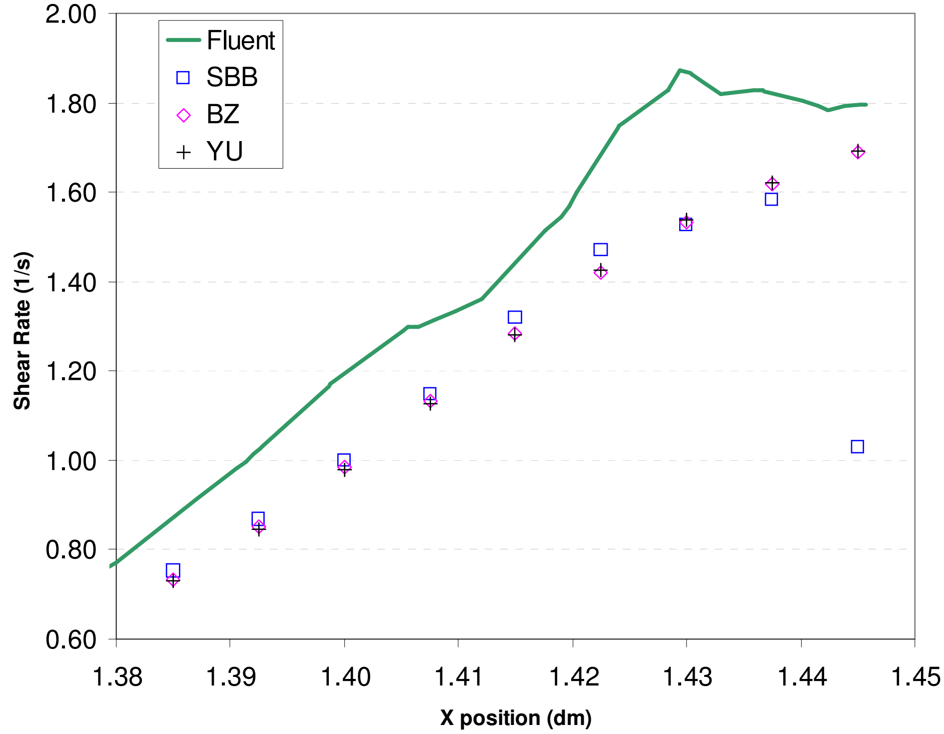


Figure 5.4 – Shear Rate Plot for the 3D Disc comparing Boundary Treatments

From the figures, the interpolation boundary treatments perform very similarly in both the 2D and 3D cases. The 2D results show better tracking of the interpolation boundary treatments to the FLUENT solution for both x-y shear stress and shear rate in comparison with the SBB treatment. For the 3D, the difference is very apparent especially at the construct surface where the shear calculations for the SBB treatment diverge significantly from both the interpolation results and the FLUENT results. Thus, while the SBB treatment may produce comparable RMS velocity accuracy over an entire main grid or sub-grid, it appears that the velocity errors at construct boundaries are high. This is most likely due to approximating the location of the boundary rather than accounting for its exact location. The result of this is that while the shear-stress and shear-rate contour plots are comparable across boundary treatments, the boundary treatment must be considered if the shear stress or shear rate is being calculated at a surface.

The rankings for the dynamics are based on the results from the 2D and 3D dynamics studies involving particle sedimentation and the motion of a neutrally-buoyant particle in an annulus. In these studies, the performance of the boundary treatments is compared. Figure 5.5 is the plot of the dimensionless drag coefficient C versus the channel geometry parameter L/d for the 2D sedimentation of a circular particle. The calculated drag coefficient, which is determined by the terminal velocity of the particle, is very similar for all the boundary treatments tested and falls in line with the analytical solution of Faxen and Takaisi.

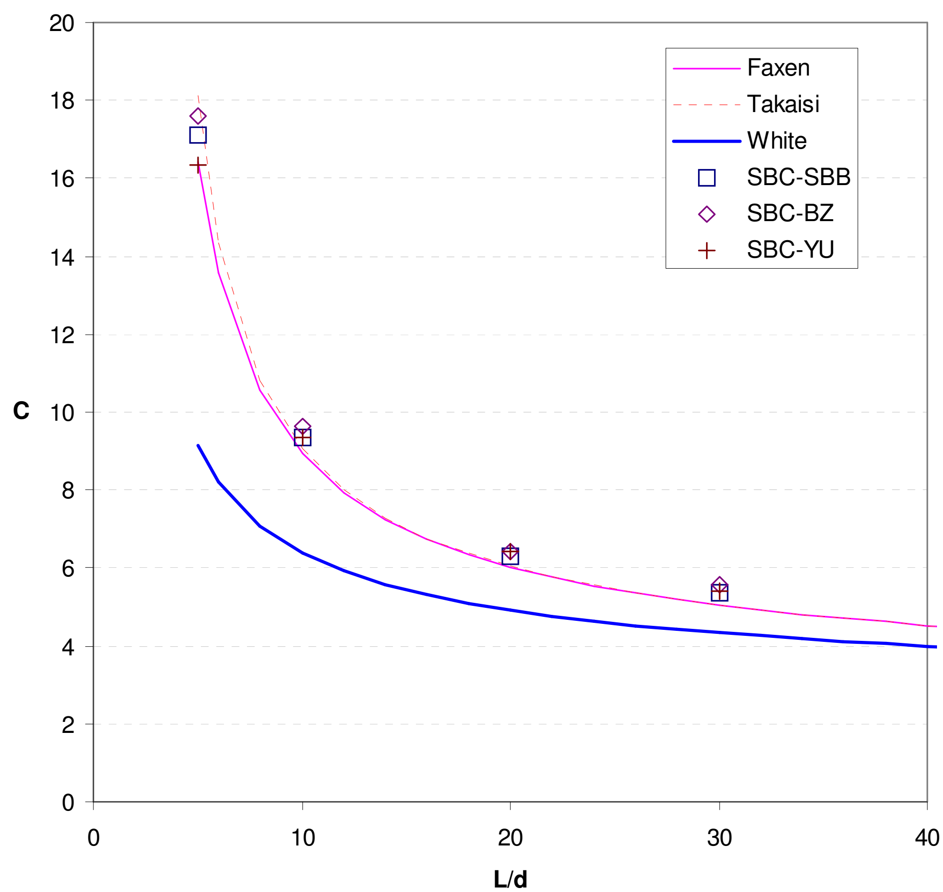


Figure 5.5 – C versus L/d comparing Boundary Treatments

Some differentiation between boundary treatments can be made when looking at the force experienced by the particle as it falls in the channel. Figure 5.6 is the comparison of forces on the particle for the different boundary treatments in the VCF3 channel geometry. By observation, the variation in forces is less for the BZ treatment compared to the SBB and YU treatments. This variation in force most likely causes the differences in particle velocity seen in Figure C.61.

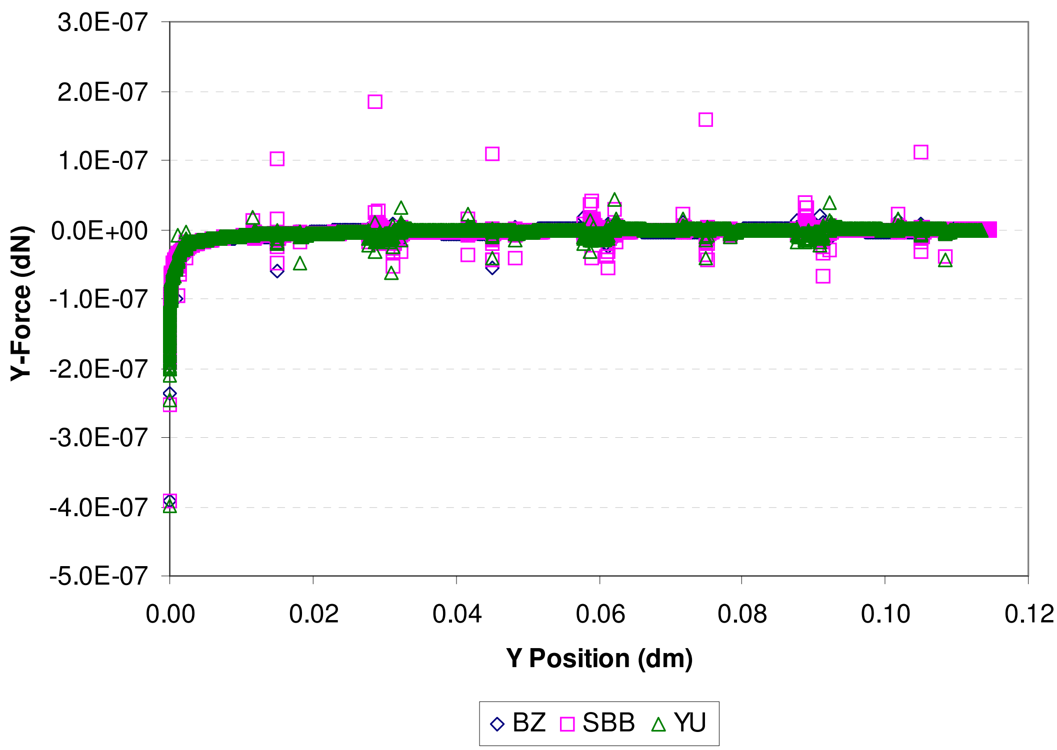


Figure 5.6 – Particle Force versus Position for the VCF3 Grid

The other 2D dynamics case tested involved the motion of neutrally-buoyant particle in an annulus. The particle paths for the circular particle are plotted for the various boundary treatments in Figure 5.7. The plot shows that for the rigid body rotation case, the lattice Boltzmann produces accurate results regardless of boundary treatment. The 3D results also indicate that both the SBB and BZ boundary treatments produce the correct particle motion in the annulus for the rigid body rotation case.

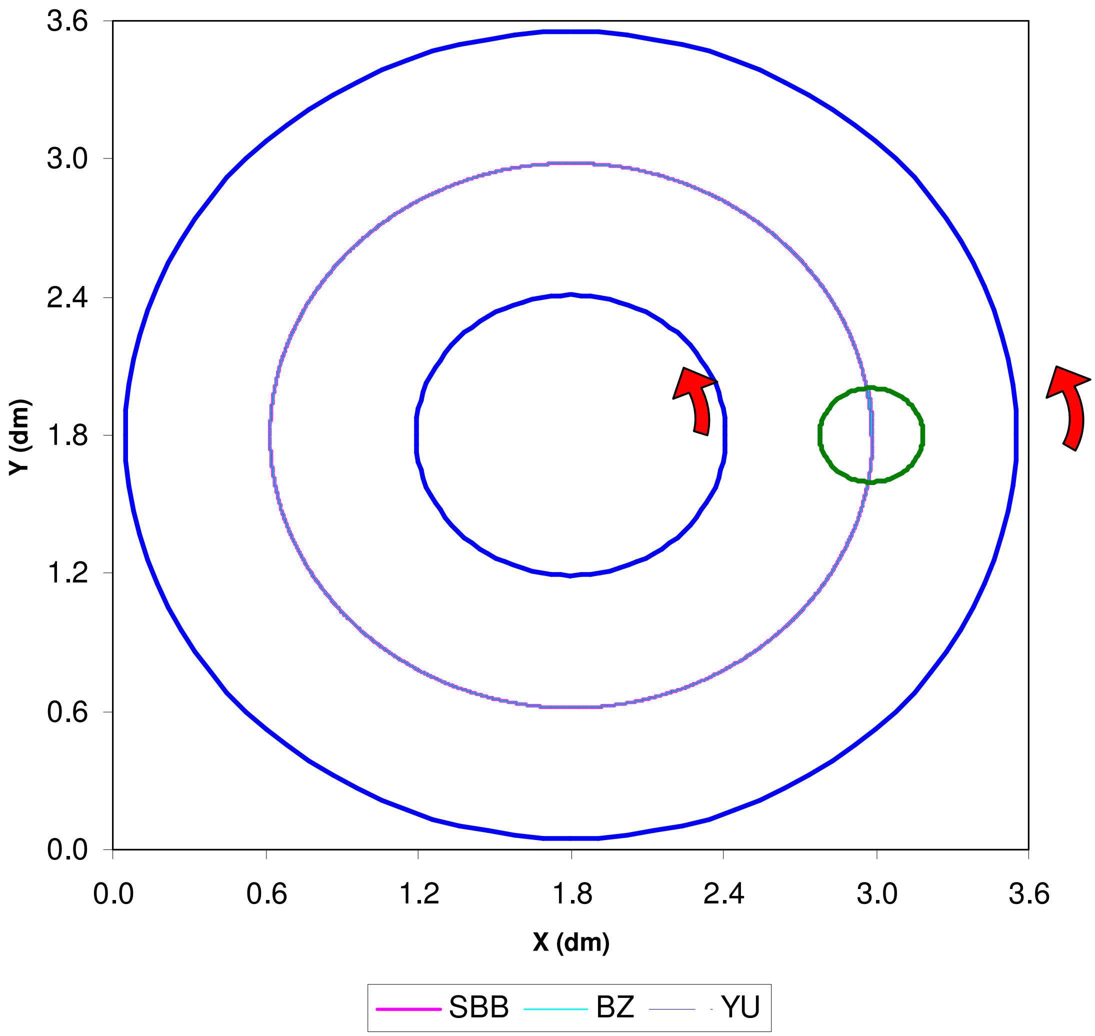


Figure 5.7 – Circle Particle Path for ANN3 Case

For the 3D particle sedimentation and motion of a neutrally-buoyant particle in a 3D annulus, a comparison across boundary treatments for the spherical case yields similar results. In the case of particle sedimentation, Figure D.71 shows that for the dimensionless constant E, which also depends on the spherical particle's terminal velocity, there is little variation across boundary treatments. The motion of a spherical particle in a 3D annulus is plotted in Figure D.77. The motion of the particle using different boundary treatments is similar to the 2D results. Again, this shows that the SBB and BZ boundary treatments perform similarly for these cases. Thus, this is reflected in the dynamics rankings for the boundary treatments.

Overall, the research presented here has allowed for a comparison of boundary treatments that examines important issues such as stability, robustness, and velocity accuracy. This knowledge should help in choosing the correct boundary treatment for particular problems in future applications. For example, the SBB treatment may not be suitable for a 2D problem, but may be adequate for a 3D moving boundary problem not involving shear stress evaluation. For the RWV bioreactor simulations, this knowledge of boundary treatments is applied so that the code performs more efficiently.

5.1.2 Grid Refinement

Methods for refining the grid in the lattice Boltzmann method have been previously proposed in the research literature. However, the current grid refinement implementation needed to be rigorously tested with respect to velocity accuracy and force evaluation accuracy. The implementation of grid refinement used here is simple and flexible in that a refined grid can be placed anywhere on the main lattice grid. The refined grid in this implementation can also be made to follow a particle as it moves in the flow. The intent of this study was to test the accuracy of the grid refinement implementation so that it can be confidently applied to various problems where it may speed up the efficiency of the code.

The grid refinement implementation was tested in two key areas: velocity accuracy and force evaluation accuracy. For testing velocity accuracy, the reduction RMS velocity error and shear stress accuracy when comparing the main grid results with sub-grid results was of primary concern. Other factors affecting velocity accuracy, such as the effect of main grid velocity accuracy, were also examined. For the force evaluation accuracy, the reduction in percent error in force or torque, with respect to FLUENT force and torque results, when comparing the main grid and sub-grid results was the main measure of effectiveness. The accuracy of the sub-grid force evaluation was also examined for the dynamics studies which involved moving boundaries.

The results for the grid refinement studies indicate that the current implementation does indeed increase the accuracy of main grid results on sub-grids. Proof of this include the reduction of absolute error on sub-grids, the reduction of velocity RMS error for sub-grids, and the reduction of error in the calculated shear stress or shear rate within the fluid. Figure 5.8 illustrates the reduction in x-velocity absolute error in going from the 2D LDCF4 main grid to the refined sub-grid located in the upper left corner of the cavity. The red arrows indicate the direction the lid is moving.

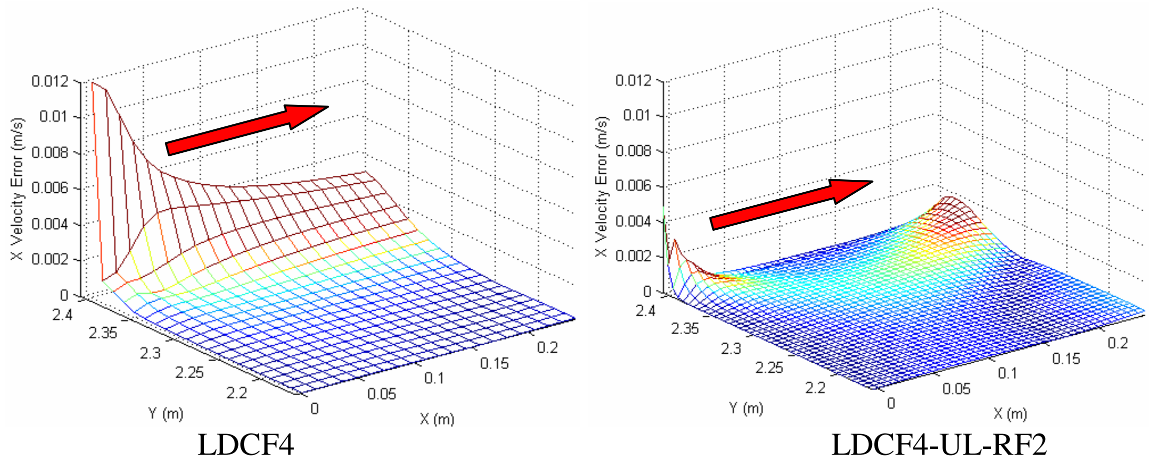


Figure 5.8 – Absolute X-Velocity Error for the LDCF4 Main Grid and Sub-Grid

The reduction of velocity RMS errors is seen for a variety of cases presented in the 2D and 3D results (See appendices C and D). Tables 5.9 and 5.10 highlight some of these results. The tables show that for a given sub-grid, the RMS error of that sub-grid is less than the RMS error for the area of the main grid covered by the sub-grid (SGA). A significant decrease in error is seen for both 2D and 3D cases and across all boundary treatments.

Table 5.9 – Percent Decrease in X-Velocity RMS Error for 2D Cases

Grid	Boundary Treatment	RMS error X-Velocity Sub-Grid (m/s)	RMS error X-Velocity SGA (m/s)	% Decrease in RMS error
LDCF4-UL RF2	SBB	0.000389	0.001614	75.92%
	MDI	0.000245	0.000666	63.23%
	BZ	0.000266	0.000625	57.51%
	YU	0.000302	0.000959	68.56%
ANN3-SQ RF4	SBB	0.000053	0.000112	52.61%
	MDI	0.000044	0.000079	44.76%
	BZ	0.000038	0.000072	46.67%
	YU	0.000042	0.000087	51.93%

Table 5.10 – Percent Decrease in Y-Velocity RMS Error for 3D Cases

Grid	Boundary Treatment	RMS error Y-Velocity Sub-Grid (m/s)	RMS error Y-Velocity SGA (m/s)	% Decrease in RMS error
LDCF3-UR RF4	SBB	0.000878	0.001152	23.76%
	BZ	0.000597	0.001324	54.88%
	YU	0.000752	0.001321	43.05%
ANN3-CUBE RF3	SBB	0.000253	0.000483	47.56%
	BZ	0.000170	0.000276	38.40%
	YU	0.000201	0.000328	38.66%

An increase in the accuracy of the shear stress calculations is also seen when grid refinement is applied. Figures 5.9 and 5.10 show the improvement that grid refinement can make in shear rate and shear stress calculations. Figure 5.9 is a shear rate line plot for the case of flow in a 2D annulus around a stationary circular construct. The sub-grid shear rate values track better than the main grid values to the FLUENT solution. Note that the construct location is approximately between X = 2.8dm and X = 3.2dm. In Figure 5.10, the x-z shear stress is plotted for case of flow in a 3D annulus around a stationary spherical construct. Again, the sub-grid shear stress values track better than the main grid shear stress values with respect to the FLUENT solution. Note that in this figure, the spherical construct is located between X = 1.4dm and X = 1.6dm. The result that these shear stress and shear rate calculations track closer with the FLUENT solution is expected because as the velocity accuracy increases on the sub-grid, so too should the accuracy of these calculations, which are based on velocity gradients.

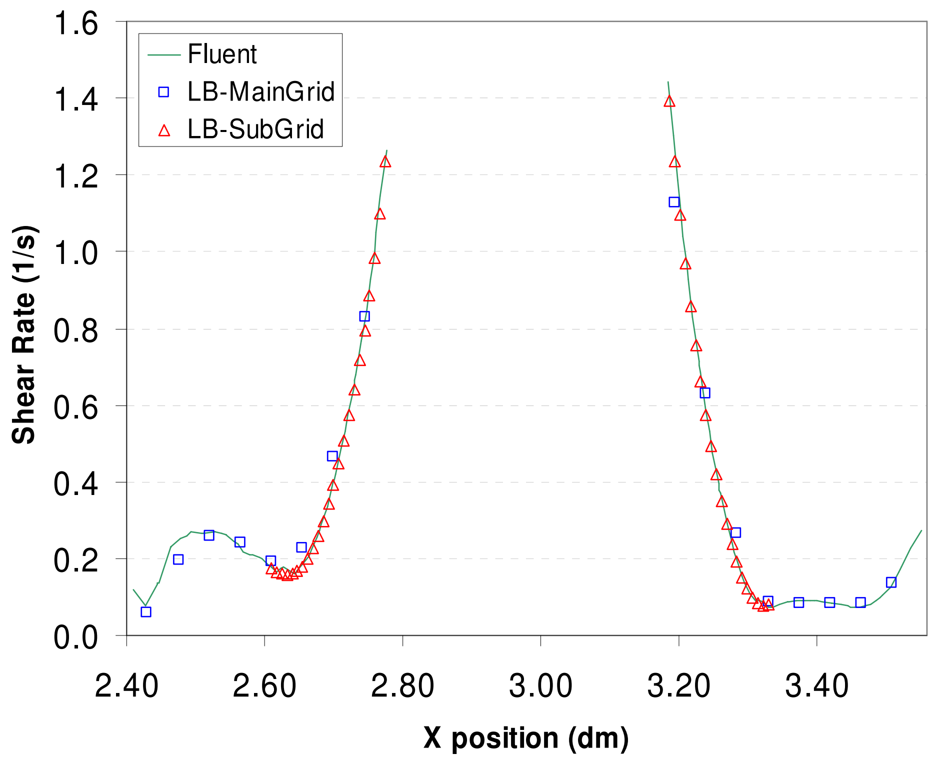


Figure 5.9 – Shear Rate Plot comparing Main Grid and Sub-Grid Values

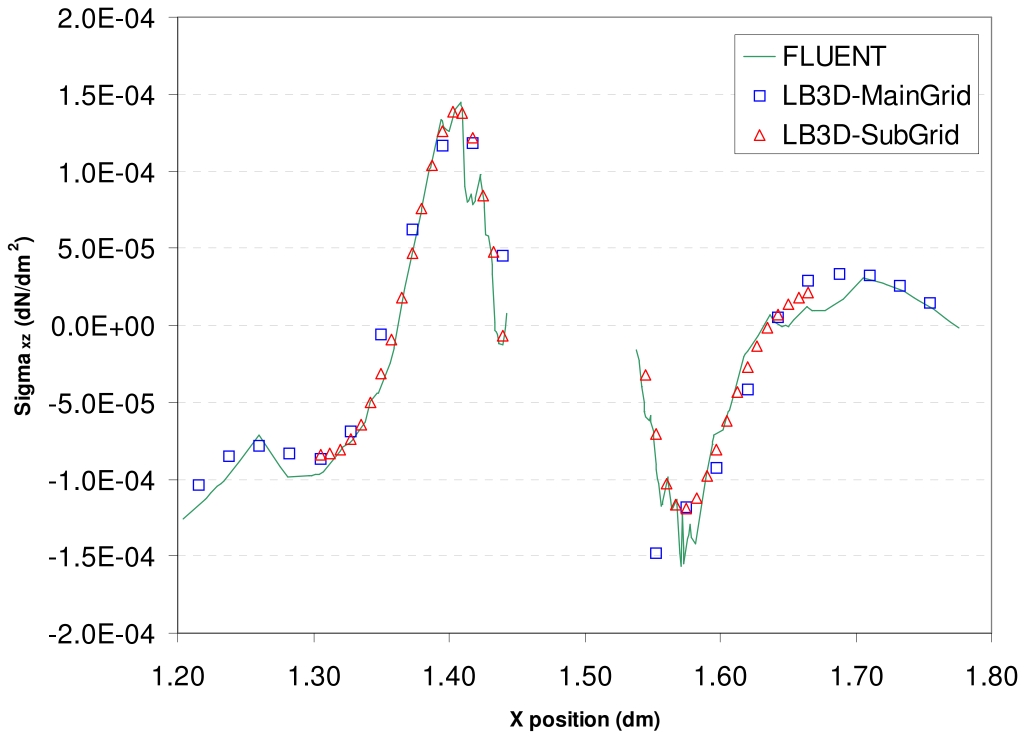


Figure 5.10 – XZ Shear Stress Plot comparing Main Grid and Sub-Grid Values

Other aspects of the grid refinement with respect to velocity accuracy were also examined. For instance, it was found that for this implementation, a refined grid containing no boundaries (i.e. there is no opportunity to increase accuracy) does not significantly degrade the accuracy of the solution. The RMS error of these sub-grids was found to be within 3% of the main grid SGA (See Tables C.17, D.21, and D.22). Another interesting result concerning velocity accuracy with grid refinement was that the velocity accuracy of the main grid did affect the accuracy of the sub-grid. In other words, sub-grids that were derived from finer resolution main grids tended to yield more accurate sub-grid solutions. Table 5.11 and Figure 5.11 illustrate this point. In Table 5.11, although all of the sub-grids have the same resolution, the RMS error trends downward as the main grid resolution increases (from ANN1 to ANN3 in this case). To further illustrate the point, Figure 5.11 is a shear stress line plot for flow around the right edge of a rectangular construct in a 2D annulus. The results for the ANN3 sub-grids tend to track closer to the FLUENT solution versus the ANN2 sub-grids.

Table 5.11 – Y-Velocity RMS Error Comparison for ANN-CIR Sub-Grid Cases

Grid	Boundary Treatment	RMS error Y-Velocity Sub-Grid (m/s)	RMS error Y-Velocity SGA (m/s)
ANN1-CIR RF12	SBB	0.000494	0.000890
	MDI	0.000365	0.000594
	BZ	0.000474	0.000669
	YU	0.000424	0.000648
ANN2-CIR RF6	SBB	0.000154	0.000295
	MDI	0.000077	0.000133
	BZ	0.000082	0.000127
	YU	0.000090	0.000142
ANN3-CIR RF4	SBB	0.000153	0.000242
	MDI	0.000061	0.000069
	BZ	0.000045	0.000044
	YU	0.000069	0.000078

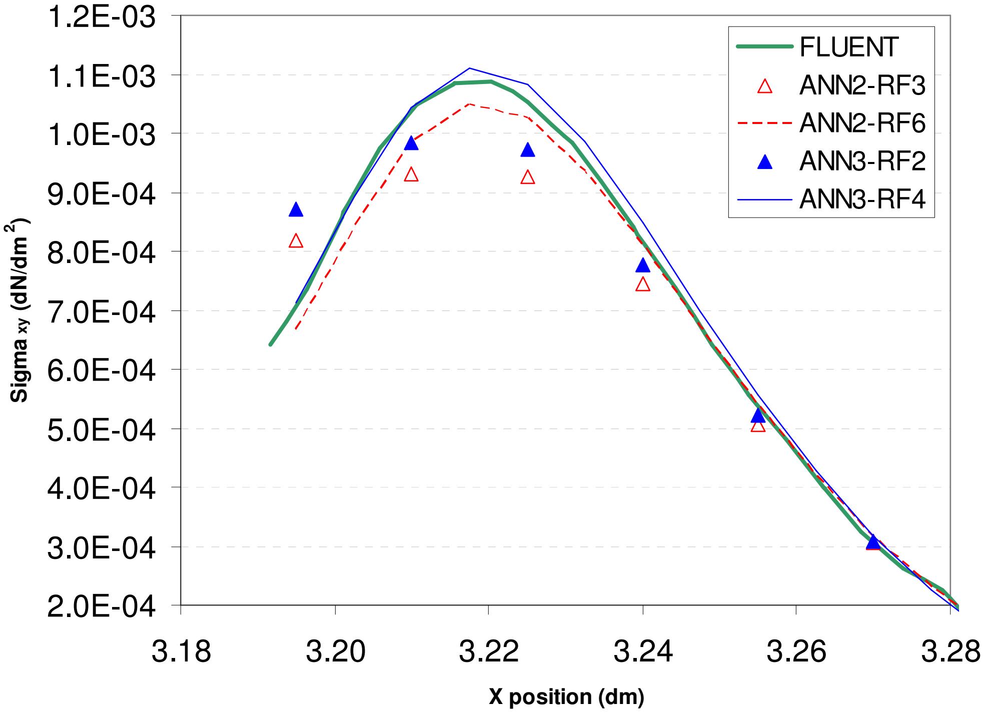


Figure 5.11 – Shear Stress Plot comparing results across Main Grids

The other key area in evaluating the grid refinement was force evaluation. This was done by examining the force results for the annulus cases involving a stationary construct, and comparing the main grid force results to the sub-grid force results. The 2D results showed that use of a sub-grid to evaluate the force yielded better approximations of the forces and torques. Table 5.12 shows the main grid force and torque results compared with the sub-grid results for flow around the ellipse construct on the ANN3 main grid.

Table 5.12 – Force Percent Error Results Comparing 2D Main Grid and Sub-Grid Cases

Grid	Boundary Treatment	X-Force Error (%)	Y-Force Error (%)	Z-Torque Error (%)
ANN3 EL	SBB	16.71%	1.26%	101.76%
	MDI	7.07%	6.21%	78.13%
	BZ	19.32%	3.86%	124.69%
	YU	22.98%	1.25%	65.31%
ANN3 EL-RF4	SBB	14.69%	0.84%	8.38%
	MDI	2.37%	1.06%	0.10%
	BZ	7.40%	2.02%	5.20%
	YU	9.02%	2.00%	18.48%

The 3D force and torque results were somewhat inconclusive based on the fact that a more refined 3D FLUENT grid may be needed for better results. While the x-direction force and y-direction force showed little to no improvement with grid refinement, there was a significant increase in z-torque accuracy. Despite this, further support for the accuracy of force evaluation for sub-grids is found by examining the results for the flow of a neutrally-buoyant particle in a 2D annulus. For the 2D results, a coarser grid of resolution 21x21 was used along with the SBB boundary treatment. It was found that after the particle was released, the computation on the main grid became unstable. This was due to the low number of nodes that resolved the construct area (the diameter of the construct was only 2 lattice spacings wide). When a sub-grid of refinement factor 6 was applied, the computation became stable and the path of the particle was as expected. This is illustrated for the 2D case in Figure 5.12 below. The particle ends up returning to its original position after being released in the rigid body rotation case. This result shows that a moving sub-grid can give correct forces and produce the correct motion of the particle.

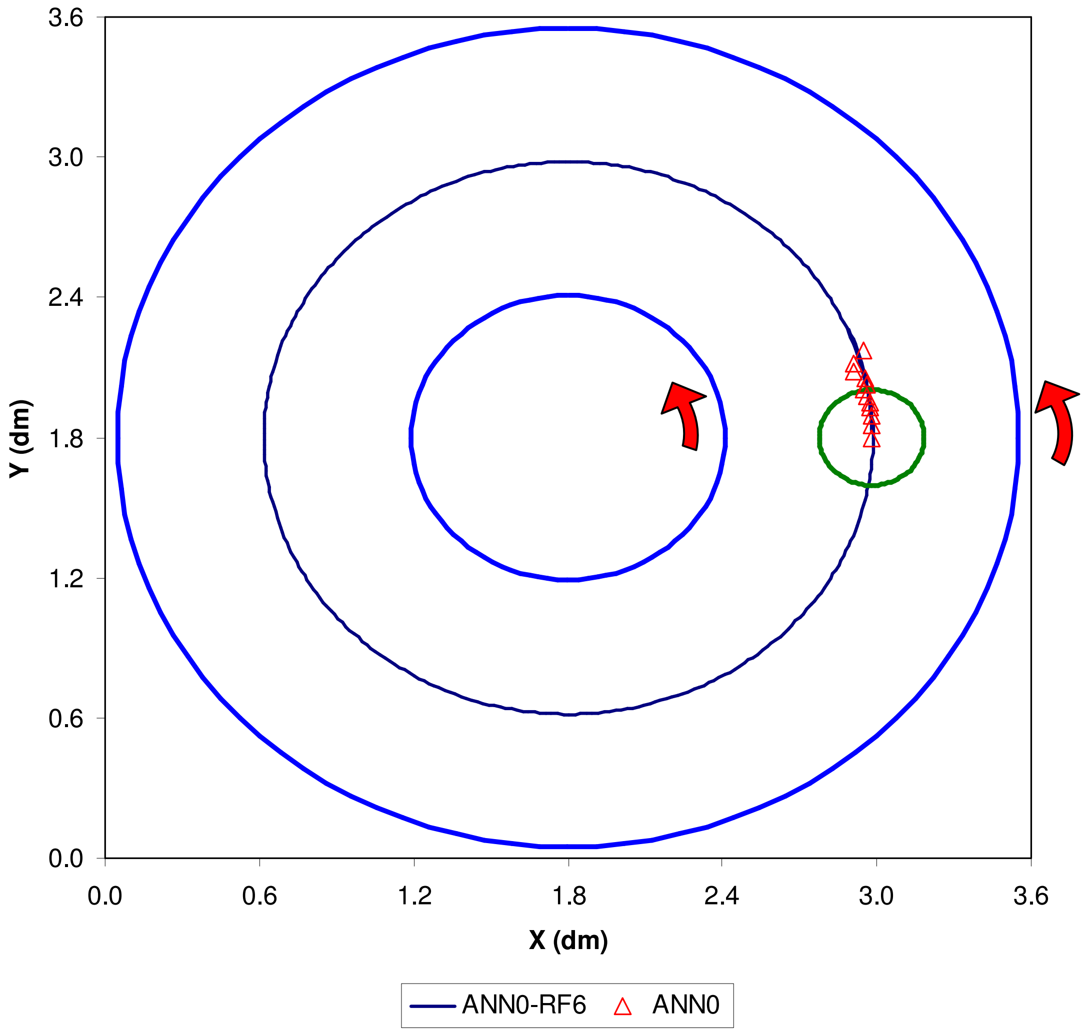


Figure 5.12 – Circle Particle Path for ANN0 Case

Overall, the grid refinement has proven to be effective in increasing the velocity accuracy and force evaluation accuracy of the lattice Boltzmann implementation. The current grid refinement implementation, which allows for a dynamic refined grid to be placed anywhere in the flow, should allow for numerous problems to be examined more effectively using lattice Boltzmann.

5.1.3 Dynamics

The dynamics studies done in this research aimed at testing the accuracy of the force evaluation, shear stress evaluation, and the handling of moving boundaries for the current implementation of the lattice Boltzmann. In order to test the accuracy of the force and shear stress evaluation, the lattice Boltzmann results were compared with results generated by the commercial CFD package FLUENT. The test cases involved stationary constructs placed in the annulus geometry. Initially, two issues came about concerning the equation to use in force evaluation. The first dealt with boundary treatments and the second involved applying a clarification to the equation previously published.

For force evaluation using the interpolation boundary treatments the method used by Aidun (Equation 2.50), which did not account for fluid inside a particle, was modified (Equation 3.1) to account for boundaries that did not exist halfway between nodes. However, the force evaluation used with the SBB treatment was the Aidun method. Using the modified momentum exchange method (or modified Aidun method) was more in-line with the interpolation boundary treatments and resulted in better force calculations in 2D. Table 5.13 is an example of matching the original Aidun force evaluation method with the interpolation boundary treatments. The table illustrates the improvement seen with using the Aidun method with interpolation. Also, note that using the modified Aidun method with the SBB treatment produces erroneous results. Thus, the modified Aidun method was used for the interpolation treatments, and the original Aidun method was used for the SBB treatment.

Table 5.13 – Force Percent Error Results for 2D Annulus Force Evaluation Cases

Grid	Boundary Treatment	X-Force Error (%)	Y-Force Error (%)	Z-Torque Error (%)
ANN3 EL-RF4 Aidun	SBB	14.69%	0.84%	8.38%
	MDI	12.93%	2.61%	101.19%
	BZ	7.94%	3.55%	104.26%
	YU	6.33%	3.54%	80.68%
ANN3 EL-RF4 Modified Aidun	SBB	33.80%	0.99%	119.01%
	MDI	2.37%	1.06%	0.10%
	BZ	7.40%	2.02%	5.20%
	YU	9.02%	2.00%	18.48%

One other issue that arose in force evaluation involved a subtle calculation clarification of the force magnitude that needed to be applied in using the original Aidun and the modified Aidun force evaluation equation. Equation 5.1 is Aidun force evaluation equation that has been published in literature.

$$\mathbf{F}(\mathbf{x}_{\mathbf{W}}, t+\frac{1}{2}\Delta t) = \frac{\Delta x^3}{\Delta t}\left[2 f_i^{\,c}(\mathbf{x}_{\mathbf{A}}, t) - \frac{2 w_i \rho(\mathbf{u}_{\mathbf{W}} \cdot \mathbf{e}_{\mathbf{i}})}{c_s^2}\right]\mathbf{e}_{\mathbf{i}} \qquad (5.1)$$

The calculation that may be overlooked when applying this equation involves evaluating this equation for lattice Boltzmann links that have a velocity magnitude of √2. For these "diagonal links", the magnitude of the force must be multiplied by the square of the velocity magnitude (from control volume analysis). Thus, a factor of 2 is needed for these links. In addition, an area correction must be made since the effective area which the force should be calculated over is no longer Δx^2 along the diagonal links, but is now (√2)/2 * Δx^2. This is illustrated in Figure 5.13 where the effective 2D area is clearly shown (assumes a unit depth of Δx). If this area correction is not applied, then the calculation is "double" counting the force or calculating force over overlapping areas. In the figure, the 2D boundary is oriented 90 degrees with respect to the diagonal link corresponding to the particle distribution f_2. The force calculated at Node BI along link f_2, should only cover the blue area of the boundary. Similarly, the forces calculated at CI and CII along f_2 should only cover the black and red areas of the boundary respectively. Even if the boundary were curved, the effective area by which the force should be calculated over is still represented by the colored segments.

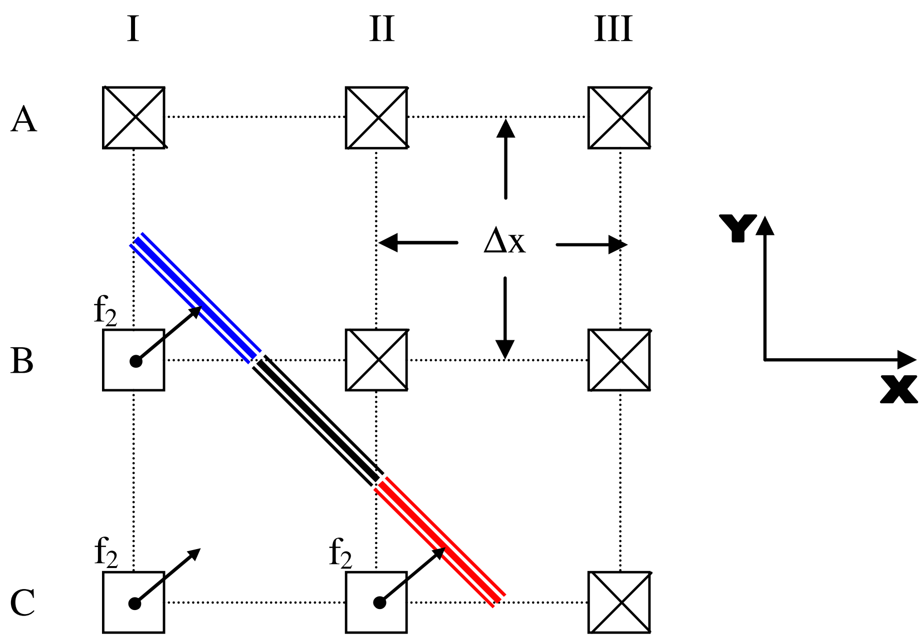


Figure 5.13 – Force Evaluation Area Illustration

So the total correction for the force along links having a velocity magnitude of $\sqrt{2}$ is multiplying by 2 for the velocity difference and $\sqrt{2}/2$ for the area difference. This clarification of the calculation applies both to the 2D and 3D lattice Boltzmann models. The clarification for the Aidun method is listed in Equation 5.2. Table 5.14 is a comparison of the percent error in force calculated with and without this clarification applied to the Aidun method and the modified Aidun method. Although only results for the 2D ellipse construct are shown, this was typically the case across all constructs and boundary conditions. Note that the direction of the force is using the unit vector of $\mathbf{e_i}$.

$$\mathbf{F}(\mathbf{x_W}, t + \frac{1}{2}\Delta t) = \frac{\Delta x^3}{\Delta t}\left[2 f_i^c(\mathbf{x_A}, t) - \frac{2 w_i \rho(\mathbf{u_W} \cdot \mathbf{e_i})}{c_s^2}\right]\hat{\mathbf{e}}_\mathbf{i} \text{ for } |\mathbf{e_i}| = 1 \tag{5.2}$$

$$\mathbf{F}(\mathbf{x_W}, t + \frac{1}{2}\Delta t) = \sqrt{2}\frac{\Delta x^3}{\Delta t}\left[2 f_i^c(\mathbf{x_A}, t) - \frac{2 w_i \rho(\mathbf{u_W} \cdot \mathbf{e_i})}{c_s^2}\right]\hat{\mathbf{e}}_\mathbf{i} \text{ for } |\mathbf{e_i}| = \sqrt{2}$$

Table 5.14 – Force Percent Error Results for 2D Annulus Clarification Cases

Grid	Boundary Treatment	X-Force Error (%)	Y-Force Error (%)	Z-Torque Error (%)
ANN3	SBB	26.25%	10.99%	17.22%
EL-RF4	MDI	16.45%	11.26%	10.76%
Clarification	BZ	20.76%	12.12%	6.55%
Missed	YU	22.64%	12.13%	26.58%
ANN3	SBB	14.69%	0.84%	8.38%
EL-RF4	MDI	2.37%	1.06%	0.10%
Clarification	BZ	7.40%	2.02%	5.20%
Applied	YU	9.02%	2.00%	18.48%

Once the correct force evaluation method for each boundary treatment was applied and the clarification was applied, the 2D force and torque results matched closely to the FLUENT results. The results that matched closest were those using the refined grids to calculate the force and torques. Although main grid calculations, which had ~10 lattice spacings across the constructs, may have given good results for one component of the force, a more refined grid gave accurate results across all forces and torques. In general, a resolution having at least ~20 lattice spacings across the construct was needed to resolve the force accurately. An example of this is in the 2D results listed in Table

5.15. In the table, the percent error for the x-direction force, y-direction force, and z-torque are given for the ANN3 main grid cases and an ANN3 sub-grid case.

Table 5.15 – Force Percent Error Results for 2D Annulus Cases

Grid	Boundary Treatment	X-Force Error (%)	Y-Force Error (%)	Z-Torque Error (%)
ANN3-CIR	SBB	32.36%	5.24%	34.50%
	MDI	2.36%	2.31%	102.32%
	BZ	40.97%	4.93%	63.12%
	YU	18.42%	1.92%	69.79%
ANN3-CIR RF4	SBB	5.19%	7.72%	10.97%
	MDI	8.46%	3.22%	2.44%
	BZ	5.87%	2.68%	4.41%
	YU	6.89%	3.49%	2.87%
ANN3-RECT	SBB	224.88%	0.03%	64.96%
	MDI	207.35%	0.17%	109.19%
	BZ	226.82%	5.00%	71.35%
	YU	426.73%	0.48%	95.37%
ANN3-RECT RF4	SBB	78.09%	7.66%	12.70%
	MDI	23.43%	5.20%	14.09%
	BZ	18.91%	5.01%	2.52%
	YU	18.47%	5.66%	8.91%

The 3D force and torque results are similar in that they trend to overall better results when grid refinement is applied. However, there was little to no improvement in the x-direction and y-direction forces when compared with the FLUENT results. The z-torque accuracy did consistently improve for all the constructs. Table 5.16 lists examples of this for the 3D annulus cases. While these results may diminish the expected effectiveness in applying a refined grid for greater for accuracy, the shear stress evaluation results and the studies involving moving boundaries do support the accuracy of the 3D grid refinement and the lattice Boltzmann for evaluating forces.

Table 5.16 – Force Percent Error Results for 3D Annulus Cases

Grid	Boundary Treatment	X-Force Error (%)	Y-Force Error (%)	Z-Torque Error (%)
ANN3-SPH	SBB	1.53%	2.41%	13.54%
	BZ	9.21%	2.15%	12.14%
	YU	7.39%	2.29%	18.21%
ANN3-SPH RF3	SBB	8.28%	11.93%	7.86%
	BZ	16.62%	10.38%	4.56%
	YU	15.68%	9.96%	0.96%
ANN3-DSC	SBB	17.69%	3.45%	74.99%
	BZ	40.92%	2.37%	144.23%
	YU	28.39%	0.97%	93.78%
ANN3-DSC RF3	SBB	21.41%	11.72%	0.00%
	BZ	31.12%	9.97%	12.63%
	YU	33.52%	8.19%	13.33%

The results from evaluating the shear stress and shear rate accuracy show that the lattice Boltzmann does well in tracking with the FLUENT results. The current implementation, especially when using grid refinement, does extremely well in 2D and fairly well in 3D. The evaluation of the current implementation was based on shear stress and shear rate plots along lines that intersected the construct. Some of these plots have already been presented in this chapter (Figures 5.1 to 5.4 and Figures 5.9 to 5.11). Figures 5.9 and 5.10 illustrate how well the calculated shear rate and shear stresses track with the FLUENT solution in both 2D and 3D. However, the 3D FLUENT results for shear stress and shear rate did appear to be coarse and not as refined as the 2D FLUENT results. This can be seen in the Figures 5.14 and 5.15. Figure 5.14 is an x-y shear stress plot at the right edge of the 3D disc construct. The jaggedness of the shear stress FLUENT solution is evident here when compared with the 2D FLUENT solution or even the lattice Boltzmann results. Despite the coarse FLUENT solution, the lattice Boltzmann solution does track with it well even at the surface of the construct.

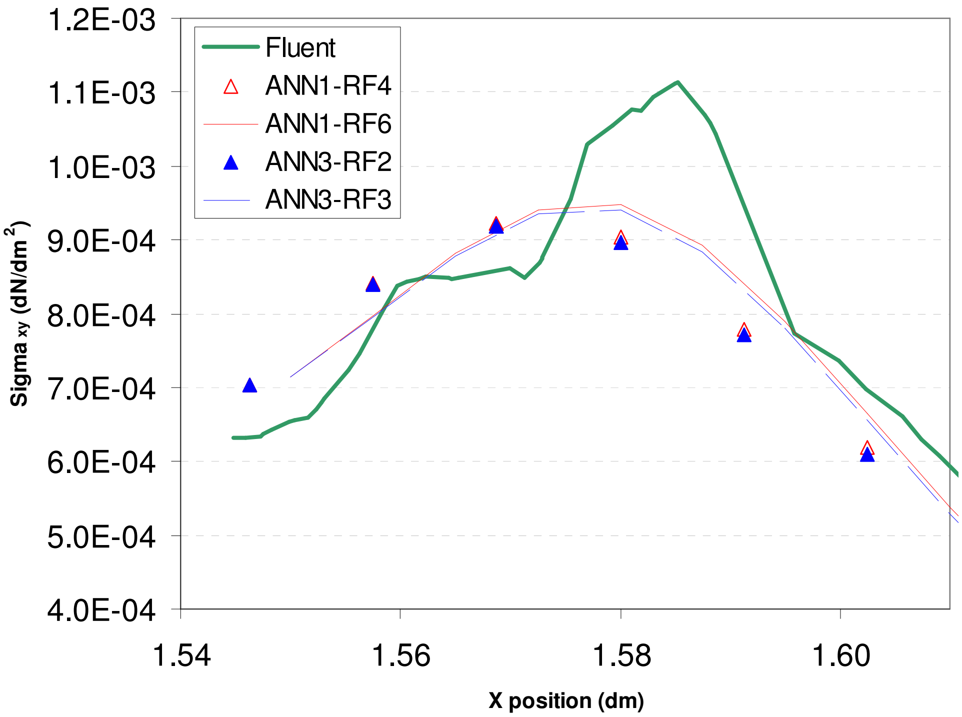
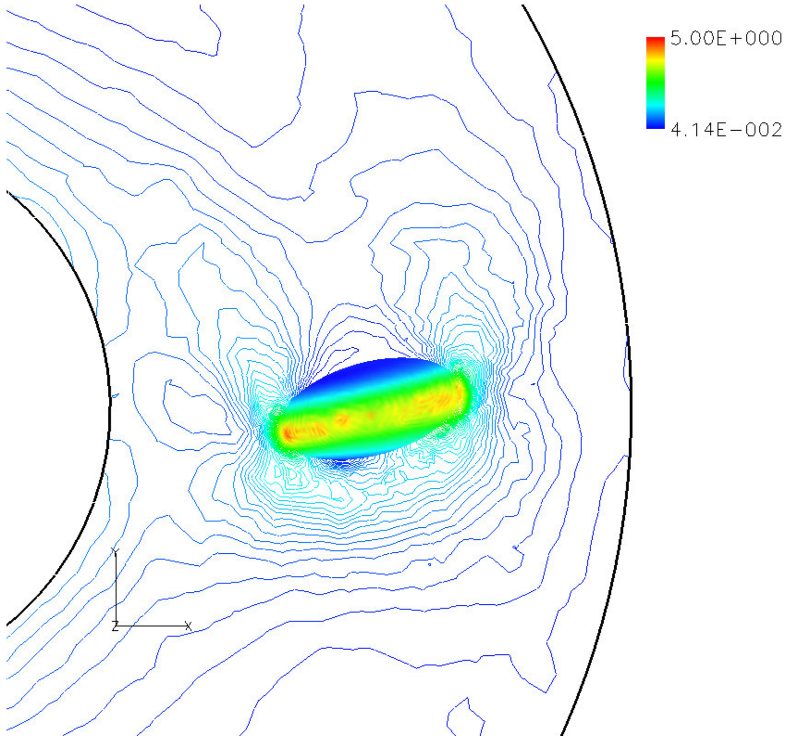
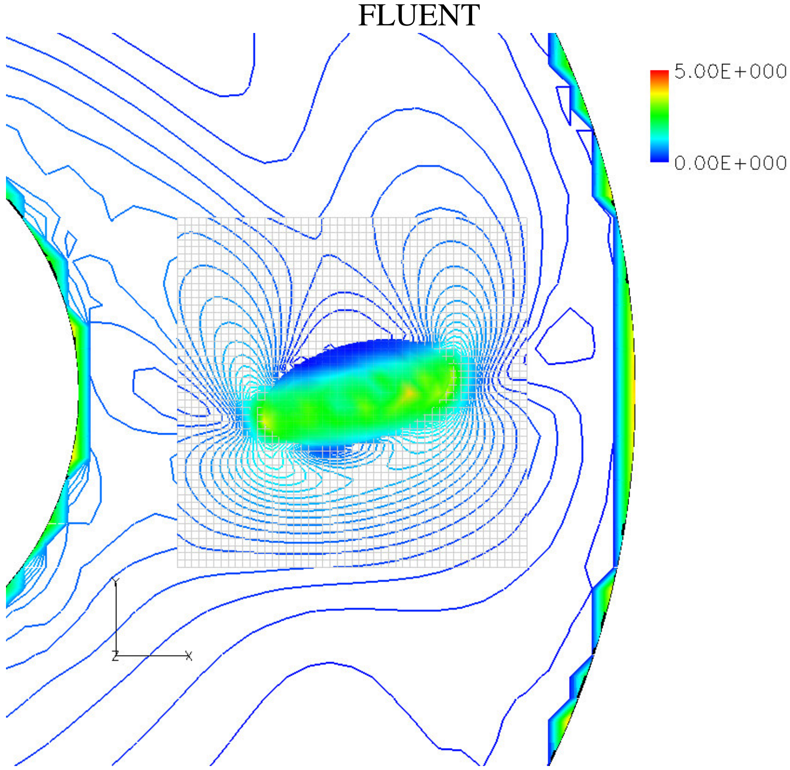


Figure 5.14 – XY Shear Stress Plot Illustrating 3D FLUENT Solution

In Figure 5.15, a side-by-side qualitative comparison of the shear rate is made using contour plots. The case analyzed here is the 3D flow around an ellipsoid construct using grid refinement. The contour plots show a good qualitative match regarding contour shape. The 3D FLUENT results do show higher shear rate at the surface of the ellipsoid surface. In examining the smoothness of the solutions, the lattice Boltzmann shear rate contours, especially within the sub-grid, are less rough than the FLUENT shear rate contours.



FLUENT



LB

Figure 5.15 – Shear Rate Contour Plots for the 3D ANN-EL Construct Case

The final cases for evaluating the dynamics methods of the lattice Boltzmann implementation involved modeling the motion of a particle. This was done through studying particle sedimentation in a 2D and 3D channel, two particle sedimentation in a 2D channel, and examining the motion of a neutrally-buoyant particle in a 2D and 3D annulus. The results of both studies provide more evidence of the accuracy of the lattice Boltzmann force evaluation and handling of moving boundaries.

For the 2D and 3D particle sedimentation, a circular cylinder or a spherical particle was dropped in a long channel and sediments under the influence of an applied force. The terminal velocity of the particle was recorded and a dimensionless constant was calculated using the terminal velocity for various channel geometries. The results of the dimensionless constants were plotted for the various channel geometries and compared with analytical solutions or experimental results. In the 2D case, the dimensionless drag constant C was plotted versus the dimensionless geometry constant L/d. Figure 5.16 are the lattice Boltzmann results for the main grids and results that used moving sub-grids centered on the particle. These results are plotted against the published analytical solutions (Faxen 1946 and Takaisi 1955) and experimental results (White 1946). From the figure it can be seen that the refined grid solutions produced a terminal velocity that is more in line with both the analytical solution and experimental results. The same holds true in the 3D results. In Figure 5.17, the dimensionless velocity constant E is plotted versus the channel geometry parameter d/L. Results from the main grids and sub-grids are plotted against the experimental results of Miyamura et al. (1981). This supports the assertion that the grid refinement does increase the accuracy of the force evaluation.

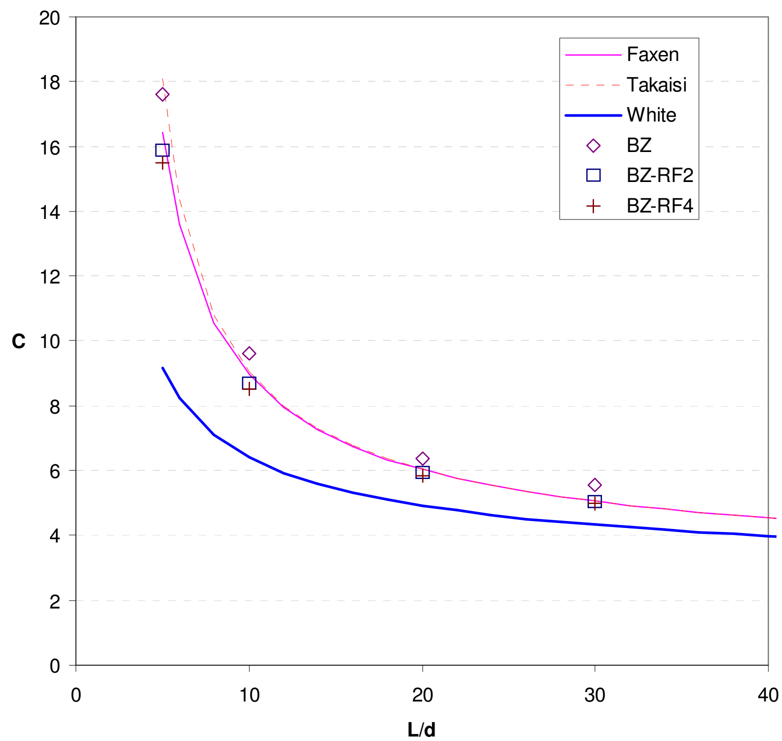


Figure 5.16 – C versus L/d with Grid Refinement for the BZ Boundary Treatment

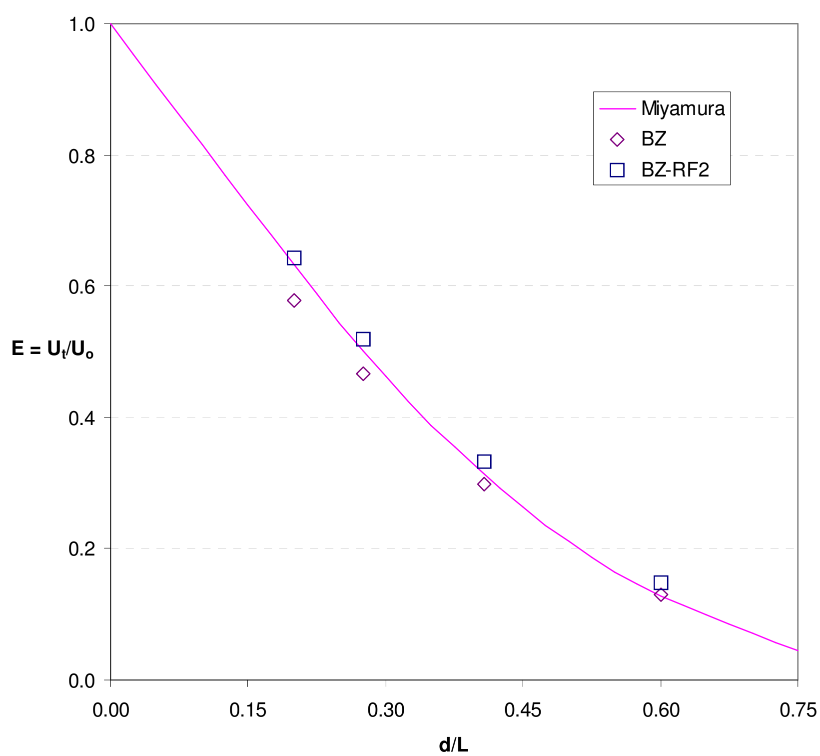


Figure 5.17 – E versus d/L with Grid Refinement for the BZ Boundary Treatment

The two particle sedimentation in a 2D channel was performed in order to further validate the lattice Boltzmann implementation. The flow parameters used were similar to those used by Aidun et al. (2003) where period doubling behavior was achieved. Figure 5.18 is a typical calculation using the SBB boundary treatment. The results obtained in these studies showed similar behavior qualitatively to both the results of Aidun et al. and Feng et al (1994). The figure shows the non-dimensional y position of the leading particle and the trailing particle versus the non-dimensional time ($t\upsilon/d^2$). The particles come in close contact and then move into a region of damped oscillation. The near contact occurs at around T = 1.2, and this is where the lubrication forces are significant. The final Reynolds number of the calculations also fell within the range of Reynolds numbers seen in the results of Aidun et al. (2003). The Reynolds numbers simulated ranged from approximately 2.8 to 4.0. In Figure 5.18, the Reynolds number is 2.96.

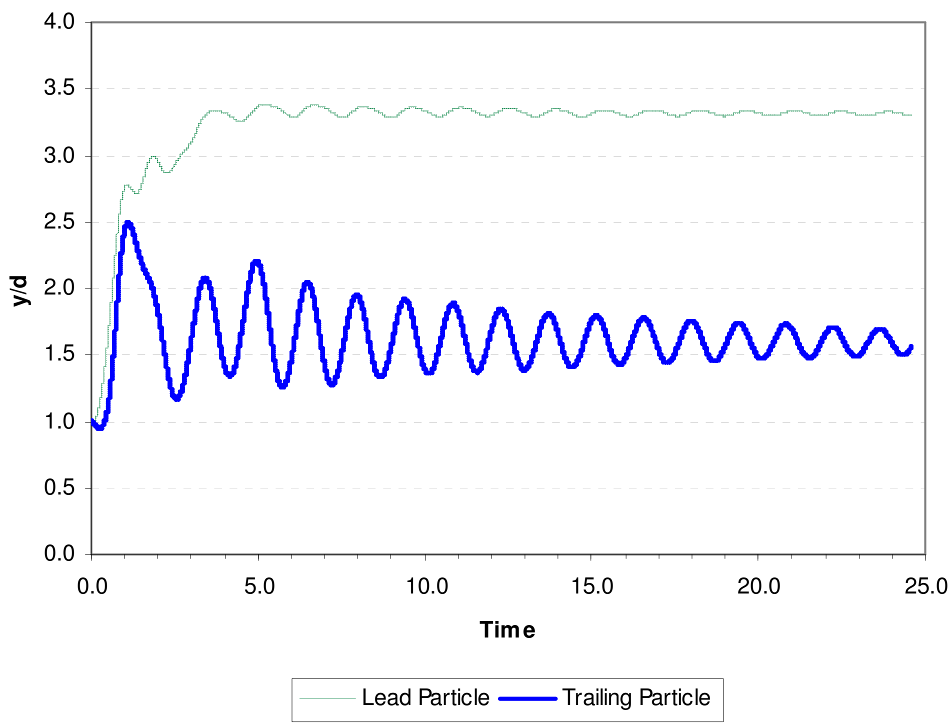


Figure 5.18 – Typical Result for Two Particle Sedimentation in a Channel

The next numerical study performed involved the movement of a neutrally-buoyant particle in a 2D and 3D annulus. The case of rigid body rotation was produced by setting the wall angular velocities. The results of this study show that motion of the particle, using forces calculated by the main grid, is as expected as the particle returns to its original position after being released. When the motion is driven by forces calculated by a moving sub-grid for a very coarse, it is also correct (See Figure 5.12). The results of the 2D annulus study were presented in Figure 5.7 and the results of the 3D annulus study are shown here in Figure 5.19. This result provides more evidence that the current implementation produces accurate forces and motion for moving boundary problems.

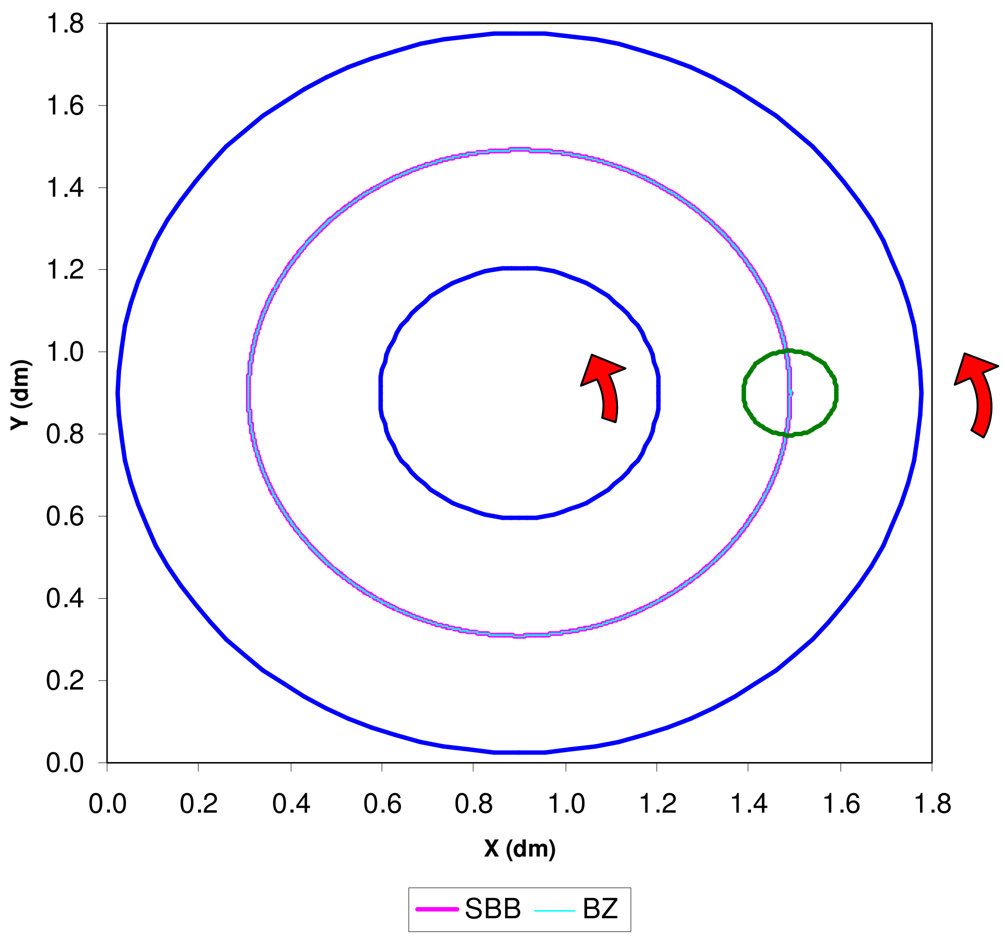


Figure 5.19 – Sphere Particle Path for 3D ANN3 Rigid Body Rotation Case

On the whole, the research presented here has verified the accuracy of the current lattice Boltzmann implementation in regards to the three areas mentioned: boundary treatments, grid refinement, and dynamics. The information presented should be useful to researchers looking to implement a basic lattice Boltzmann code to solve a variety of problems. The problem that is being modeled in the current research is the RWV bioreactor. In order to accurately simulate the basic aspects of this problem, the verification of the lattice Boltzmann implementation with regard to these three areas is crucial. The next step is then applying the current lattice Boltzmann implementation in an attempt to validate its use for the RWV bioreactor problem. The primary means in validating the method are checking if the motion of the constructs and flow structure around the constructs are the reasonably close to experimental results, and if the model gives reasonable estimates of the shear stresses near the construct surface.

5.2 Application to the RWV Bioreactor

Validation in applying the current lattice Boltzmann implementation to the RWV bioreactor problem involved simulating the motion of the construct in both 2D and 3D bioreactor geometries. The information obtained in verifying the accuracy of the lattice Boltzmann is used to choose the best flow parameters and implementation methods with which to approach the bioreactor problem. The results of the simulations are evaluated by comparing the construct motion, construct wake structure, and shear stress near the construct surface with results from previous experiments. Success in generating comparable results to experiments based on these criteria validates the use of this lattice Boltzmann code.

5.2.1 Implementation

There were two main considerations made in applying the lattice Boltzmann to the bioreactor problem: i) which lattice Boltzmann methods suit the problem best; and ii) whether the lattice Boltzmann simulation can match the flow parameters of the physical problem. The determination of which lattice Boltzmann components are used was based on practicality, dynamics accuracy, and shear stress accuracy. The matching of the physical flow parameters involved varying grid resolution and lattice Boltzmann parameters until stability of the code could be achieved.

For the 2D simulations, the most practical lattice Boltzmann implementation for this problem should use the moving refined grid technique in order to save computation time. Thus, a coarse main grid and a refined grid centered on the construct were utilized. This also allows for more accurate dynamic construct motion and more accurate velocities near the construct. The boundary treatment chosen for the 2D simulations was the SBB treatment. This was based on its 2D performance regarding dynamics accuracy for the moving sub-grid case. The MDI treatment was not chosen because of time constraints, but the BZ treatment was used for 2D shear stress evaluation. For the 3D simulations, the moving refined grid technique was also chosen for the same stated reasons. The boundary treatment chosen for the 3D simulations was again the SBB treatment. However, the SBB treatment was only used on the sub-grid for its dynamics accuracy, while the BZ treatment was again used for the more accuracy concerning shear stress calculations. This was done for practicality and stability reasons.

The geometry of the RWV bioreactor problem was taken from a previously published study by Neitzel et al. (1998). Most of these parameters were closely approximated by the lattice Boltzmann computation for both 2D and 3D. The rectangular construct was chosen for the 2D simulations and the disc construct was chosen for the 3D simulations. The flow parameters that were chosen for the lattice Boltzmann simulations were based on initial steady state solutions done in order to test stability. It was found that, in both 2D and 3D, the exact kinematic viscosity could not be modeled without resulting in an unstable result. Geometric and wall velocity variation were attempted in order to match an appropriate Reynolds numbers, however the same instability occurred. Thus, the geometric parameters and wall velocities were preserved while the kinematic viscosity was raised slightly to maintain stability. Table 5.17 lists the key geometry and flow parameters for the 2D, 3D, and experimental studies.

Table 5.17 – Lattice Boltzmann Parameters for the Simulations

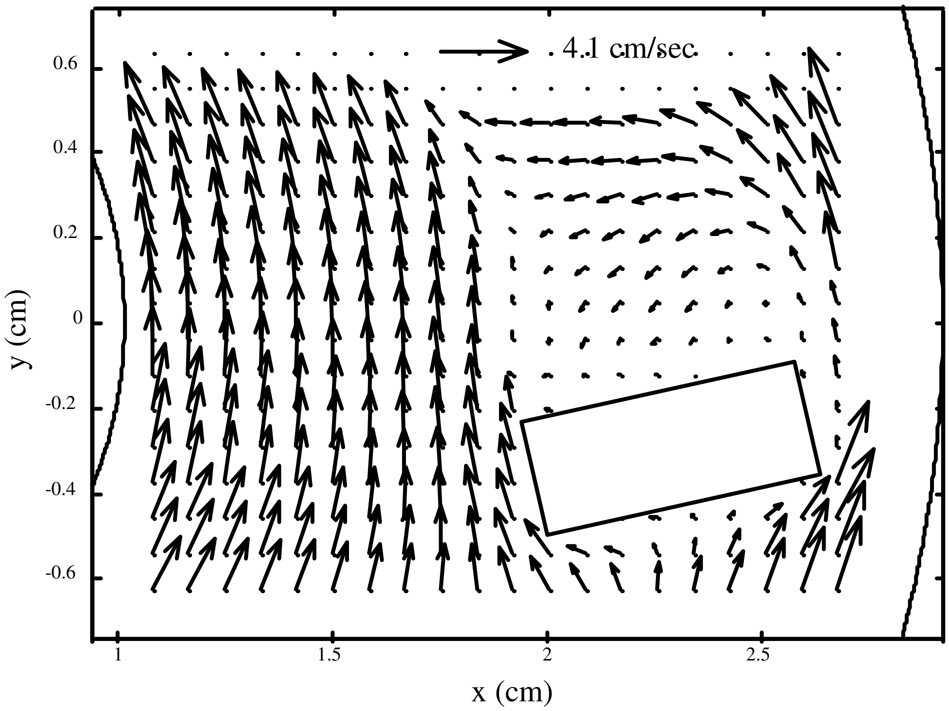
Parameter	2D LB Simulation	3D LB Simulation	Neitzel et al. (1998)
Inner Wall Radius	0.1015dm	0.1015dm	0.1dm
Outer Wall Radius	0.291dm	0.291dm	0.29dm
Inner Wall Velocity	0.4dm/s ~ 13 RPM	13-14 RPM	Varied (E) 13 RPM (C)
Outer Wall Velocity	0.4dm/s ~ 37 RPM	13-37 RPM	Varied (E) 37 RPM (C)
Construct Radius	0.034dm	0.03205dm	0.02875-0.03285dm
Fluid Kinematic Viscosity	$0.0005dm^2/s$	$0.0005dm^2/s$	$0.0001dm^2/s$ = 1cS
Fluid Density	$0.816kg/dm^3$	$0.816kg/dm^3$	$0.816g/cm^3$
Construct Density	$0.830kg/dm^3$	$0.830kg/dm^3$	$0.830g/cm^3$
Grid Resolution	81x81	81x81x81	N/A

E – Experiments C – Computations

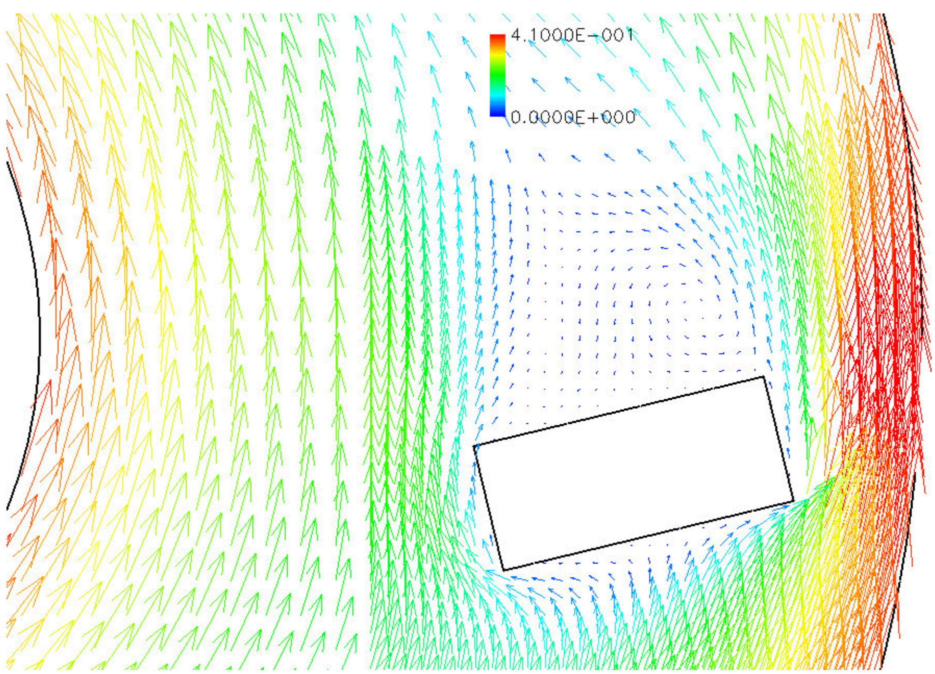
5.2.2 Simulations

Although the real RWV bioreactor involves flow in 3D, 2D simulations were performed. The 2D simulations were performed in order to see if the construct behavior described in experiments could also be reproduced in 2D. The results from the 2D simulations show that while the wake structure varied from the experimental results, some of the same construct behavior seen in experiments could be demonstrated in the 2D simulation. This behavior included the construct that mainly stayed stationary in the 3 o'clock position ("floating") and the orbiting of the construct around a fixed point. In addition, shear stress calculations from 2D simulations showed reasonable values around 1.0 to 2.0 dynes/cms^2. However, while the 2D results were encouraging, the 3D simulations results would be a better measure of the validation of the lattice Boltzmann code.

For the 3D simulations, the results again compared favorably with experiments. This was the case in comparing the wake structure, the construct behavior, and the estimated shear stresses. First, the wake structure of the 3D flow past the "floating" construct matches that of previous computational and experimental results by Neitzel et al. (1998). Figure 5.20 is a comparison of the wake seen in the lattice Boltzmann simulation with the wake visualized by PIV in the Neitzel study. Note that the lattice Boltzmann velocity vectors are colored by velocity magnitude and that the scale is in dm/s. From the figure comparison, it can be seen that the wake behind the construct is similar, especially for the area of recirculation at the right edge of the construct. However, there are differences seen in velocity magnitudes. In the experiments, the velocity magnitude is higher which in part is due to the kinematic viscosity difference. This difference in velocity is also seen in Figure 5.21, where a velocity line plot at Y = -0.375cm shows higher PIV experiment values for velocity.



Neitzel et al. (1998)



LB

Figure 5.20 – Velocity Vector Comparison for the 3D Flow around a DSC Construct

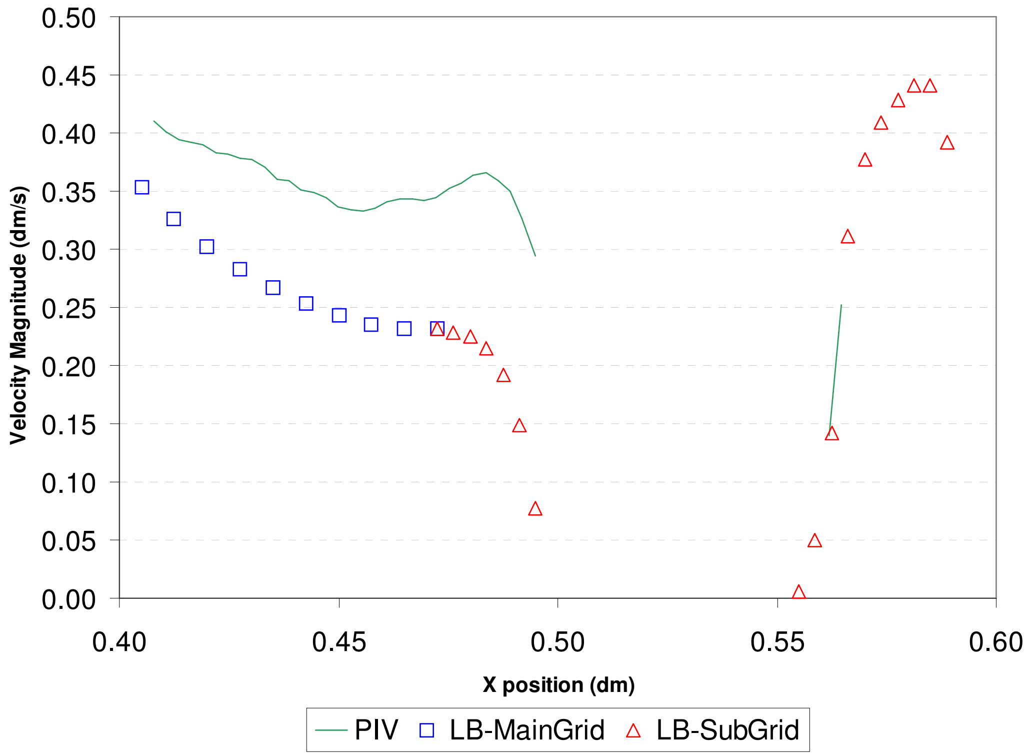


Figure 5.21 – Line Plot at Y = -0.375cm for Velocity Magnitude Comparison between PIV and LB data

The dynamic motion of the disc construct in 3D also compared favorably to what had been seen experimentally. The construct was observed "floating" in place at 3 o'clock, orbiting about the annulus, or orbiting about some fixed point in the annulus. The most interesting case is the case where the construct orbits about some fixed point. In Figure 5.22, the construct path is plotted for the case where both inner and outer wall annulus velocities were set to 14RPM, with the red arrow showing the direction of its movement. It is interesting that the circular orbit of the construct in the 3D simulation matches the shape of the orbit seen previously by Brown (1998). While the radius of the orbit here may not be as large as other results, the reproduction of the behavior is validation of the dynamics calculations done by the lattice Boltzmann implementation.

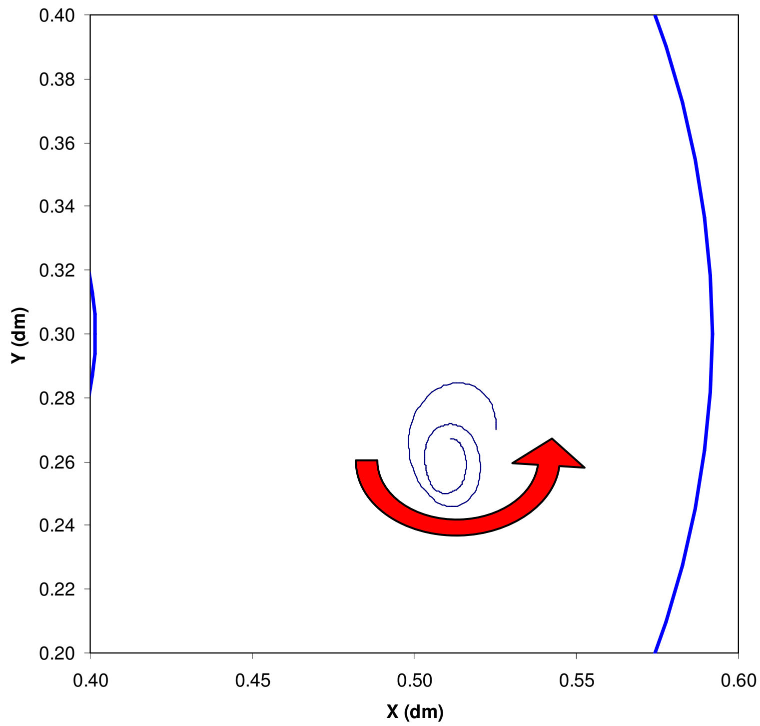


Figure 5.22 – DSC Construct Path for the 14RPM Case

The shear stress results for the 3D simulations also support the validity of lattice Boltzmann implementation. Although the kinematic viscosity was not precisely matched in the lattice Boltzmann simulations, the estimated shear stresses calculated were in the range of the expected shear stresses. For the case of the construct orbiting about a fixed point, the maximum x-y shear stress observed in the fluid was approximately 2.78 dynes/cm^2. A second estimation of the average shear stress on the construct for the 3D simulations was 1.75 dynes/cm^2. These values are close to the 1.5 dynes/cm^2 value estimated by Freed and Vunjak-Novakovic (1995) and the 0.8 dynes/cm^2 value estimated by Neitzel et al. (1998). Thus, the current lattice Boltzmann implementation yields reasonable results for the shear stress.

By and large, the 3D simulations have shown that the lattice Boltzmann implementation has given reasonable results relating to the structure of the flow, the dynamic construct behavior, and the estimated shear stresses on the construct. Thus, the implementation has great potential if further studies are conducted on the RWV bioreactor problem. The results also support the use of the lattice Boltzmann to study other problems that may involve moving boundaries or particles.

5.3 Summary

In this chapter, the accuracy of the current implementation of the lattice Boltzmann was verified. It was found that in the areas of boundary treatments, grid refinement, and dynamics that this implementation performs very well. Subsequently, the application of that implementation was validated for the RWV bioreactor problem. The motion of the construct, the estimated shear stresses, and structure of the flow around the construct inside the annulus seen in experiments were reproduced in the simulations. In the process of verification and validation a number of interesting outcomes and results concerning the lattice Boltzmann have been presented. The concluding chapter will summarize the contributions of this research and recommend directions for future work.

Chapter 6

Contributions and Recommendations

This chapter summarizes the contributions made by this research and the recommendations for future work using the current lattice Boltzmann implementation. The accuracy of the current model was tested and verified against the commercial package FLUENT, experimental results, and analytical results. Some of the more subtle details of the implementation and how they contributed to the results are presented. The flexibility of the current implementation allows it to be applied to numerous other fluid flow problems. In addition, the current implementation can be augmented such that other aspects of problems like heat or mass transfer can be modeled. Recommendations on the direction of this research are included to illustrate the potential of the method.

6.1 Contributions of this Research

The contributions of this research can be grouped into two different types: the contribution to the body of knowledge in understanding and clarifying the capabilities of the lattice Boltzmann and the contribution of new implementation methods that help to advance the lattice Boltzmann method itself. Ultimately, these contributions to the lattice Boltzmann will help in applying the method to a range of problems that may have been difficult to simulate using other computational methods.

The first set of contributions involves adding to the body of knowledge concerning the abilities of the lattice Boltzmann method. Here is a summary of those contributions:

- There have been many published results concerning the lattice Boltzmann method that have showed the method as accurate. The research presented here reinforces the accuracy of the lattice Boltzmann through a testing framework that directly compares lattice Boltzmann results to results from a commercial CFD package, FLUENT. This is unique and differs from almost all of the literature available on lattice Boltzmann. The testing framework allows for the examination of summary measures of performance, like velocity RMS error, or the specific visualization of error within the geometry of the problem.
- Numerous boundary treatments in the lattice Boltzmann method have been proposed over the past decade. This research compares the performance of the simplest and most widely accepted boundary treatment, the shifted bounce-back treatment, with the performance of boundary treatments based on interpolation. In contrast to previous work, the current research tested the boundary treatments for practicality, stability, robustness, velocity accuracy, force evaluation accuracy, shear stress evaluation accuracy, and dynamics. Thus, a contribution to a comparative view of boundary treatments is made.
- Several methods for grid refinement had been previously presented in the lattice Boltzmann literature. The method implemented in this research had

been tested qualitatively with respect to velocity errors in previously published work. This research provided results that the grid refinement does decrease velocity RMS error values when implemented. This occurred for both the lid-driven cavity case and the stationary construct in an annulus case. Results also showed that force and shear stress calculations done using the refined grid were accurate compared to coarser main grids. Thus, the implementation of the grid refinement was validated and should provide computational savings for future lattice Boltzmann calculations.

- The implementation of a moving boundary in the lattice Boltzmann has been previously published by several researchers. The current implementation of the moving boundary and the dynamics involved with a moving particle was benchmarked in this research by three test cases. Previously, only the 3D sedimentation in a channel and 2D sedimentation of two particles had been done computationally using the lattice Boltzmann. The 2D sedimentation of a circular cylinder in a channel, the 2D motion of a neutrally-buoyant circular particle in an annulus, and the 3D motion of a neutrally-buoyant spherical particle in an annulus have not been previously performed. The positive results from these three problems further reinforce the suitability of the lattice Boltzmann for moving boundary problems.
- In reviewing the published literature, the link between the parameters in the lattice Boltzmann simulation and the physical parameters of a problem

may be difficult to understand for the reader. In this research, the physical parameters are assigned to the grid, which in turn help define the parameters in the simulation. This enabled the same physical problem to be modeled in FLUENT in order to have a direct comparison of results. While often dimensionless lattice units are used in published literature, the methodology described in this study should help clarify the link between the physical parameters and the lattice Boltzmann parameters.

The second set of contributions deals with the new methods that are specific to the current lattice Boltzmann implementation and help improve its performance. Here is a summary of those contributions:

- A multi-dimensional interpolation (MDI) boundary treatment was proposed and tested for the 2D lattice Boltzmann implementation. The results show that it performed similar to the other interpolation boundary treatments tested, and outperformed them in some cases.
- The grid refinement implementation was adapted so that it could be applied to a moving particle. A refined sub-grid could be placed anywhere on the main lattice grid at any time step and provide accurate results. The current implementation allows for the sub-grid to follow the particle as it travels. In addition, the forces that the sub-grid calculates can be used to drive the motion of the particle.
- In dealing with the momentum exchange rules for boundaries that translate, calculating the momentum exchange that occurs between the

particle distribution function that hits the boundary and the boundary itself involves a density term. The published literature has used the average flow density in this term. In contrast, results in this study indicate that the local node density should be used for that momentum exchange term.

- For force evaluation on a surface, the momentum exchange method proposed by Aidun (no fluid on the inside of the particle) had been previously used by researchers regardless of boundary treatment implemented. It was found in this study that a modified momentum exchange method based on interpolation should be used for boundary treatments which do not assume the boundary exists half way between nodes. However, for the shifted bounce-back boundary treatment, the original method for force evaluation proved to work well. Additionally, a clarification is mentioned for links that have a velocity magnitude of $\sqrt{2}$.
- The boundary condition used at the inlet and outlet of the 2D and 3D channel for the particle sedimentation study has not been used in previous work. The free boundary condition or Stokes boundary condition utilizes the initial equilibrium particle distribution function to fill the incoming unknown distributions. The results using the Stokes boundary condition were comparable to the results using the cavity boundary condition, but allowed for velocity to develop at the inlet and outlet.
- The primary components for evaluating the shear stress or shear rate within a fluid are the velocity gradients. While these gradients can be calculated from the lattice velocity results, a method to calculate the

gradients from the particle distribution functions themselves is presented. Both results from both methods compare favorably to each other and to FLUENT results.

6.2 Recommendations for Future Work

There are several recommendations that can be made given the research presented in this thesis. These recommendations include revising the current lattice Boltzmann implementation for better efficiency, performing more numerical studies that further test the implementation, integrating new components of the lattice Boltzmann into the current implementation for future, and applying the current work to real life problems like the RWV bioreactor to a greater extent.

- In the next major revision of this lattice Boltzmann code a few things need to be considered:
    - The program should be more streamlined with regard to memory storage. This would allow for larger 3D grids to be used.
    - The interpolation routines used in the MDI boundary treatment need to be faster. The multi-dimensional interpolation served as a bottleneck for the calculation.
    - True parallelization of the code may increase performance, especially for 3D cases.
- The following are implementation methods that need to be further tested:
    - The 3D implementation of the MDI boundary treatment needs to be thoroughly tested. Initial indication show that the 3D

performance is similar to the 3D performance of the BZ boundary treatment. However, testing using 3D cases similar to the ones presented could verify its performance.

  - Further testing on the moving sub-grid implementation could be done on more complex flow situations.

- These are some lattice Boltzmann components that could be added to the current implementation to widen the range of problems that could be investigated:
  - Heat transfer
  - Mass transfer
  - Multi-phase flow
  - Multiple particle simulations
  - Deformable boundaries

- Lastly, a more extensive application of this code to the RWV bioreactor problem or problems similar to it would be desirable. The basic functionality of the code has been proven and can be applied to problems with wide-ranging geometries. Specifically, problems where shear stress is a concern or the motion of a boundary or particle is of interest would be suitable candidates.

Appendix A

Quaternion Math

A quaternion is a four element vector normalized to unit length that can represent a 3 x 3 rotation matrix. Quaternions are less susceptible to numerical drift because they use one extra variable in describing three degrees of freedom for rotation (Witkin and Baraff 1997). This results in a reduction of the total number of equations for rigid body motion. Quaternions also allow for smooth and continuous rotations without the worry of a problematic gimbal lock. Hence, quaternions have become widely used in programming simulations and computer graphics. Figure A.1 shows one type of notation for a quaternion **q**.

$$\mathbf{q} = [S, \mathbf{V}] = [S, V_x, V_y, V_z]$$

Figure A.1 – Quaternion Notation

A quaternion can be thought of as a complex number, but is actually an example of a general class of hypercomplex numbers. The parameter S is a real part while V_x, V_y, and V_z are coefficients for the imaginary part. Typically **V** is expressed as (V_x, V_y, V_z) or $V_x\mathbf{i} + V_y\mathbf{j} + V_z\mathbf{k}$, where $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1$. The following brief examples using quaternions cover some basics of quaternion math.

A general rotation θ about an axis **u** can be represented by the quaternion [cos(θ/2), sin(θ/2)**u**]. For composite rotations, the multiplication of quaternions needs to be defined. As an example, if $\mathbf{q_1}$ and $\mathbf{q_2}$ represent rotations, then $\mathbf{q_1 q_2}$ represents the composite rotation of $\mathbf{q_2}$ followed by $\mathbf{q_1}$. Figure A.2 illustrates quaternion multiplication.

$$\mathbf{q_1 q_2} = [S_1, \mathbf{V_1}][S_2, \mathbf{V_2}] = [S_1 S_2 - \mathbf{V_1} \cdot \mathbf{V_1}, S_1 \mathbf{V_2} + S_2 \mathbf{V_1} + \mathbf{V_1} \times \mathbf{V_2}]$$

Figure A.2 – Quaternion Multiplication

The process by which a rotation matrix is converted to a quaternion and the reverse operation is needed. Typically, an initial rotation must be converted to a quaternion, the new rotation is then calculated by solving the ODE in Equation 3.20, and at each time step the current rotation matrix needs to be calculated. The rotation matrix is necessary for computations such as the inertia tensor calculation in Equation 3.18. As a side note, in Equation 3.20, the notation **Ω**(t)**q**(t) represents the multiplication of [0, **Ω**(t)] and **q**(t). The conversions are shown in Figures A.3 and A.4.

Further information on quaternions is available. The extensive notes "An Introduction to Physically Based Modeling" by Witkin and Baraff (1997) at Carnegie Mellon University are excellent. Two particular websites that have also been useful are listed.

Gamedev.net: http://www.gamedev.net/reference/articles/article1095.asp

Wolfram: http://mathworld.wolfram.com/Quaternion.html

$$\mathbf{q}(t) = \mathbf{R}(t) = \begin{pmatrix} 1-2V_y^2-2V_z^2 & 2V_xV_y-2SV_z & 2V_xV_z+2SV_y \\ 2V_xV_y+2SV_z & 1-2V_x^2-2V_z^2 & 2V_yV_z-2SV_x \\ 2V_xV_z-2SV_y & 2V_yV_z+2SV_x & 1-2V_x^2-2V_y^2 \end{pmatrix}$$

Figure A.3 – Quaternion to Matrix Conversion

```
if (trace(R(t) >= 0)
{
        r = sqrt(trace(R(t)) + 1)
        S = 0.5 * r
        r = 0.5 / r
        Vx = (R32 – R23) * r
        Vy = (R13 – R31) * r
        Vz = (R21 – R12) * r
}
else if (R11 > R22 AND R11 > R33)
{
        r = sqrt(R11 – R22 – R33 + 1)
        Vx = 0.5 * r
        r = 0.5 / r
        Vy = (R12 + R21) * r
        Vz = (R31 + R13) * r
        S = (R32 – R23) * r
}
else if (R22 > R11 AND R22 > R33)
{
        r = sqrt(R22 – R11 – R33 + 1)
        Vy = 0.5 * r
        r = 0.5 / r
        Vz = (R23 + R32) * r
        Vx = (R12 + R21) * r
        S = (R13 – R31) * r
}
else if (R33 > R11 AND R33 > R22)
{
        r = sqrt(R33 – R11 – R22 + 1)
        Vz = 0.5 * r
        r = 0.5 / r
        Vx = (R31 + R13) * r
        Vy = (R23 + R32) * r
        S = (R21 – R12) * r
}
```

Figure A.4 – Matrix to Quaternion Conversion

Appendix B

Partial Derivative Calculation

The calculation of $\partial u/\partial x$ using lattice Boltzmann PDFs and finite difference formulas starts at Equation 3.30. Breaking this equation down further, Equation B.1 can be easily derived.

$$\frac{\partial u}{\partial x} = c\frac{\partial}{\partial x}\left(\frac{f_1}{\rho}+\frac{f_2}{\rho}+\frac{f_8}{\rho}-\frac{f_4}{\rho}-\frac{f_5}{\rho}-\frac{f_6}{\rho}\right) \qquad \text{(B.1)}$$

Since the PDFs and the density can be considered functions that depend on the spatial position, the quotient rule is required in differentiating further. Using the quotient rule, the following equation is the result.

$$\frac{\partial u}{\partial x} = \frac{c}{\rho^2}\begin{pmatrix} \rho\frac{\partial f_1}{\partial x} - f_1\frac{\partial \rho}{\partial x} + \rho\frac{\partial f_2}{\partial x} - f_2\frac{\partial \rho}{\partial x} + \rho\frac{\partial f_8}{\partial x} - f_8\frac{\partial \rho}{\partial x} - \\ \rho\frac{\partial f_4}{\partial x} + f_4\frac{\partial \rho}{\partial x} - \rho\frac{\partial f_5}{\partial x} + f_5\frac{\partial \rho}{\partial x} - \rho\frac{\partial f_6}{\partial x} + f_6\frac{\partial \rho}{\partial x} \end{pmatrix} \qquad \text{(B.2)}$$

The partial derivative $\partial u/\partial x$ of a velocity value has now become a function of the partial derivative of PDFs and the density. These partial derivatives using PDFs and density are calculated using finite difference formulas, similar to those used in Equations 3.27 to 3.29. The only difference is the use of PDFs and density instead of velocity. Equations B.3 and B.4 are examples using a central and forward differences.

$$\frac{\partial f_1}{\partial x} = \frac{f_1(\mathbf{x}+\mathbf{e_1}\Delta x,t) - f_1(\mathbf{x}-\mathbf{e_1}\Delta x,t)}{2\Delta x} + O(h^2) \qquad \text{(B.3)}$$

$$\frac{\partial \rho}{\partial x} = \frac{-3\rho(\mathbf{x},t)+4\rho(\mathbf{x}+\mathbf{e_1}\Delta x,t)-\rho(\mathbf{x}+2\mathbf{e_1}\Delta x,t)}{2\Delta x} + O(h^2) \qquad \text{(B.4)}$$

Equation B.3 is a central difference in x to find the partial derivative of the PDF f_1. Equation B.4 is a forward difference in x to find the partial derivative of ρ the density. Similar equations are used for other partials. For example, the calculation of $\partial v/\partial y$ in Equation 3.31 can be further broken down into Equation B.5, which contains other partials including $\partial f_3/\partial y$ and $\partial \rho/\partial y$.

$$\frac{\partial v}{\partial y} = \frac{c}{\rho^2}\begin{pmatrix} \rho\frac{\partial f_2}{\partial y} - f_2\frac{\partial \rho}{\partial y} + \rho\frac{\partial f_3}{\partial y} - f_3\frac{\partial \rho}{\partial y} + \rho\frac{\partial f_4}{\partial y} - f_4\frac{\partial \rho}{\partial y} - \\ \rho\frac{\partial f_6}{\partial y} + f_6\frac{\partial \rho}{\partial y} - \rho\frac{\partial f_7}{\partial y} + f_7\frac{\partial \rho}{\partial y} - \rho\frac{\partial f_8}{\partial y} + f_8\frac{\partial \rho}{\partial y} \end{pmatrix} \tag{B.5}$$

Equations B.2 and B.5 are used in contrast to straight finite difference approximations such as Equations 3.27. Again the reason for this alternate method is for a method for verification and to test the smooth and differentiable assumption of PDF surfaces.

Appendix C

2D Results

The majority of the tables and figures that comprise the results from the 2D numerical studies that are not included in the body of this thesis are listed in this appendix. The intent is to provide a more complete record of the data gathered for this study. The tables and figures are referenced in the 2D results section of Chapter 4. This appendix is organized by numerical study.

Poiseuille Flow with shifted boundaries

<u>RMS Error plots for HCF Cases</u>

Note: The results for the SBB case are plotted on all 4 plots as a reference

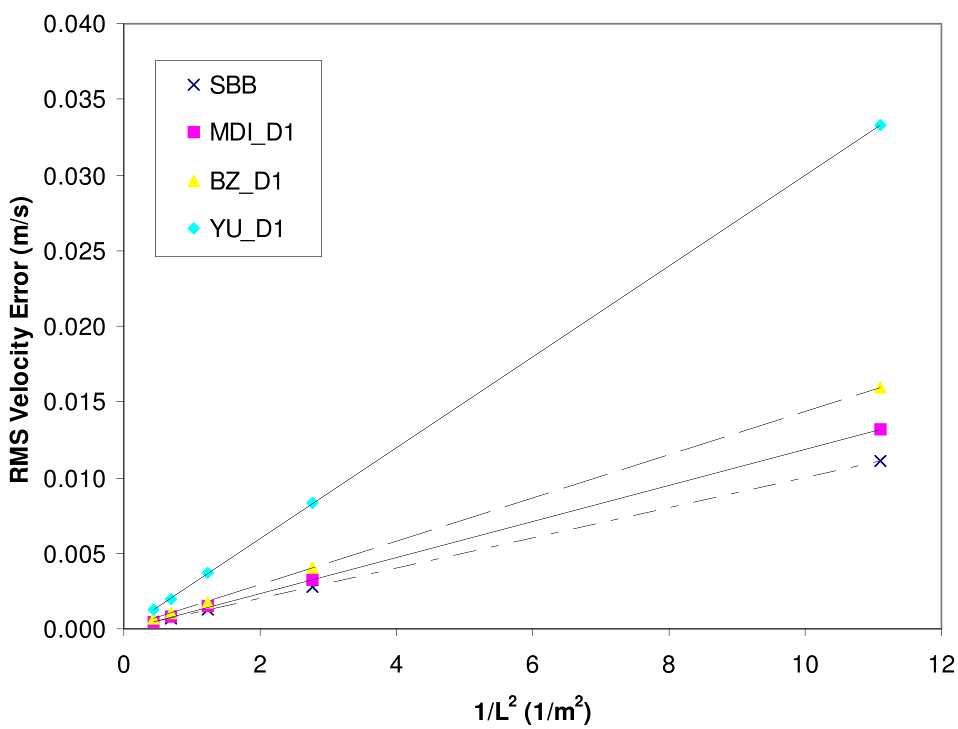


Figure C.1 – RMS Error versus $1/L^2$ for the D1 case

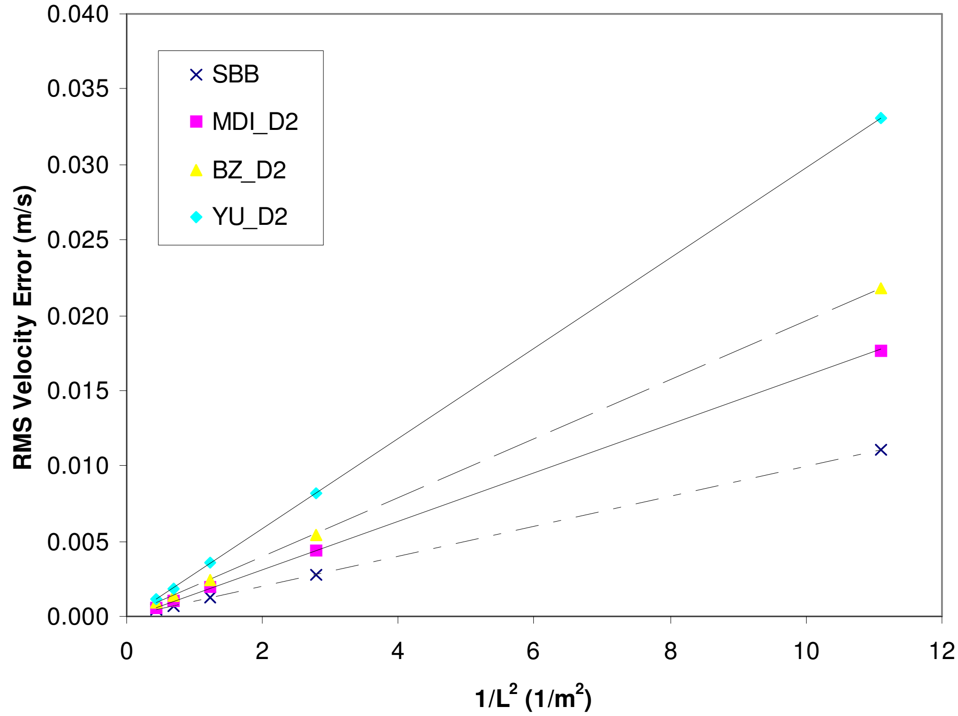


Figure C.2 – RMS Error versus $1/L^2$ for the D2 case

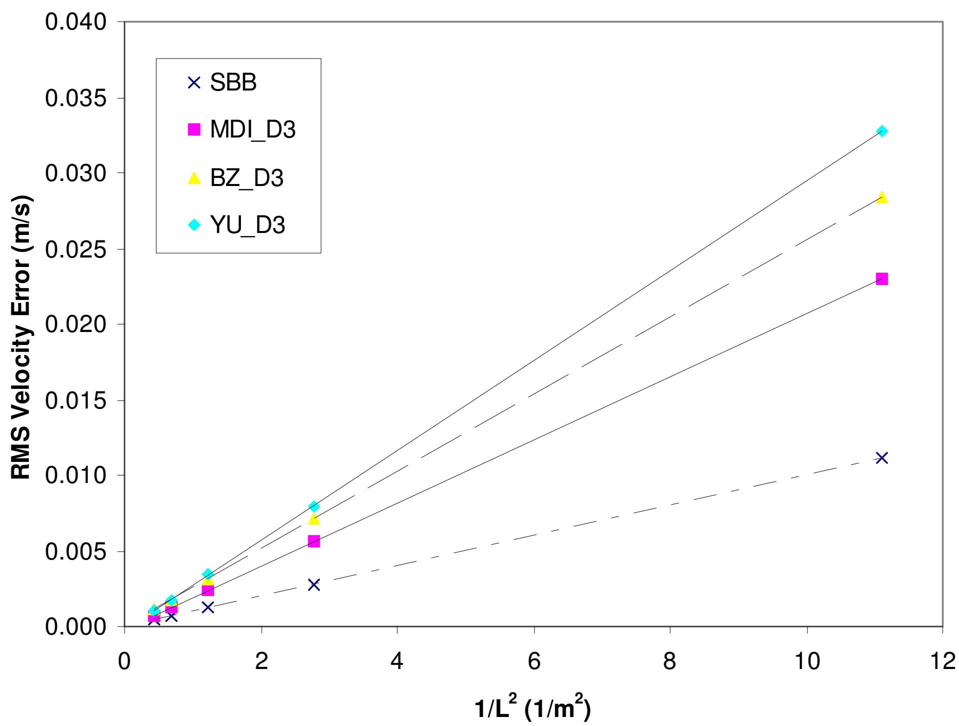


Figure C.3 – RMS Error versus $1/L^2$ for the D3 case

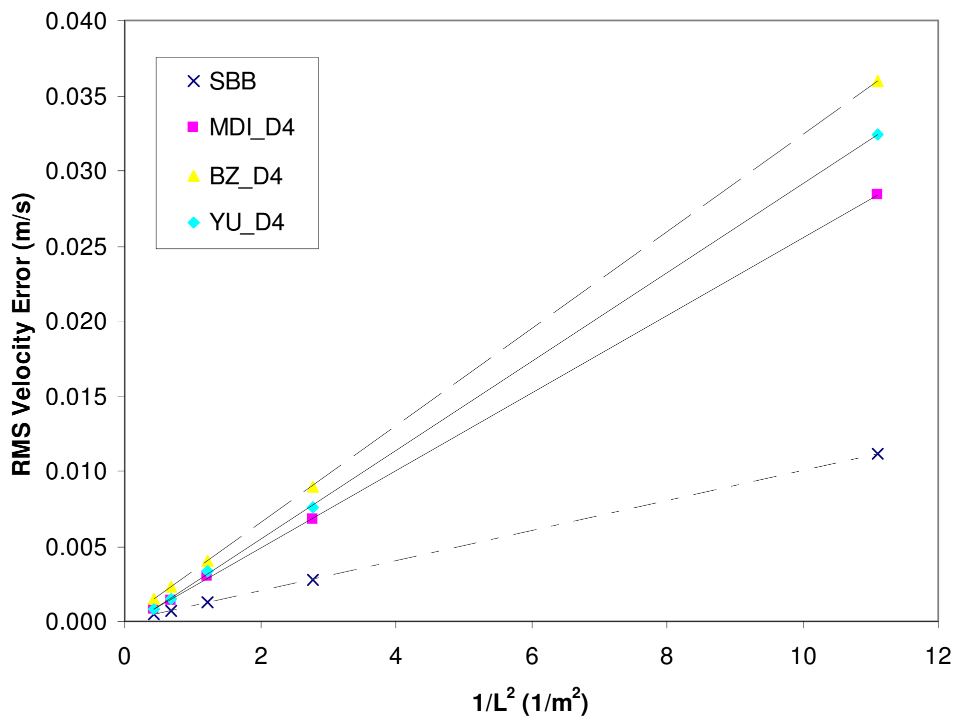


Figure C.4 – RMS Error versus $1/L^2$ for the D4 case

Absolute Error Visualization for HCF1 Case

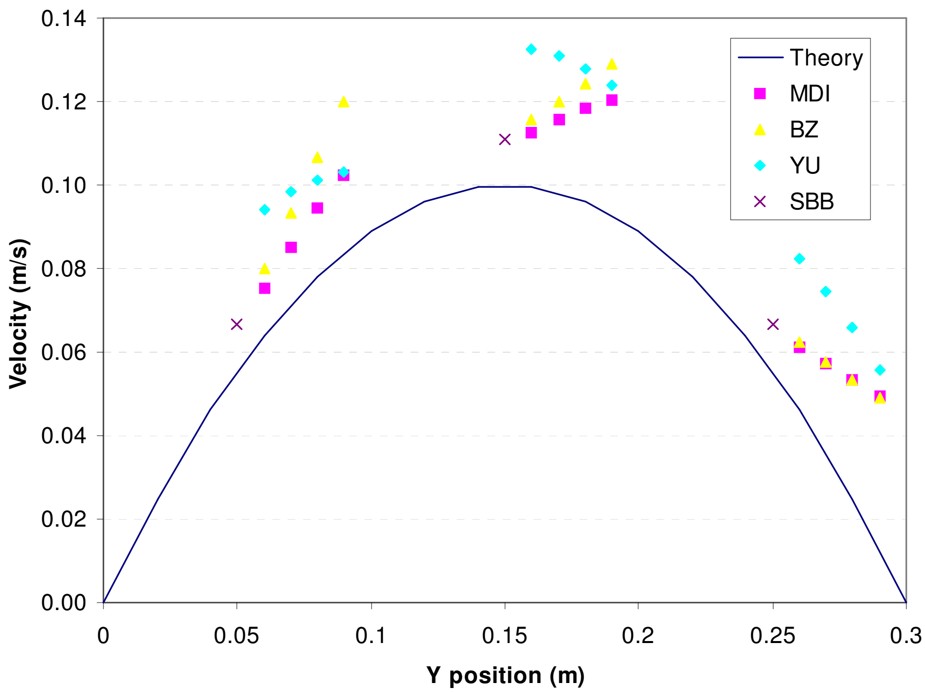


Figure C.5 – Velocity Profile for the HCF1 Grid where L = 0.3m

RMS Error Results for HCF Cases

Table C.1 – Channel Resolution (L) versus RMS Error for all Cases

		RMS Error (m/s)					
		D1 case			D2 case		
L (m)	SBB	MDI	BZ	YU	MDI	BZ	YU
0.3	0.01111	0.01314	0.01600	0.03327	0.01769	0.02178	0.03310
0.6	0.00278	0.00328	0.00400	0.00828	0.00436	0.00544	0.00814
0.9	0.00123	0.00145	0.00178	0.00368	0.00193	0.00242	0.00360
1.2	0.00069	0.00077	0.00100	0.00199	0.00098	0.00136	0.00187
1.5	0.00044	0.00048	0.00064	0.00125	0.00059	0.00087	0.00114
		D3 case			D4 case		
L (m)	SBB	MDI	BZ	YU	MDI	BZ	YU
0.3	0.01111	0.02293	0.02844	0.03281	0.02837	0.03600	0.03244
0.6	0.00278	0.00558	0.00711	0.00790	0.00676	0.00900	0.00758
0.9	0.00123	0.00245	0.00316	0.00346	0.00295	0.00400	0.00329
1.2	0.00069	0.00121	0.00178	0.00170	0.00141	0.00225	0.00151
1.5	0.00044	0.00071	0.00114	0.00100	0.00081	0.00144	0.00084

Lid-Driven Cavity Flow

Streamline Visualizations for LDCF Cases

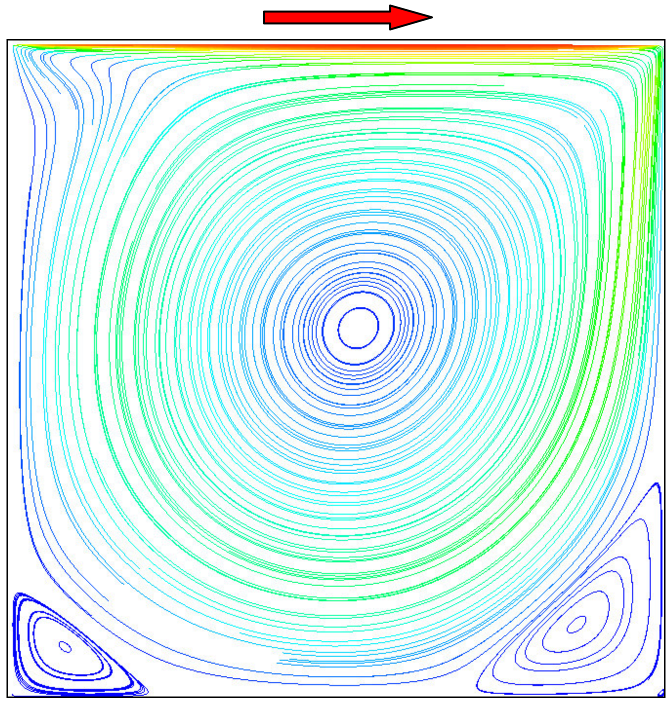


Figure C.6 – Streamlines for the FLUENT results for LDCF

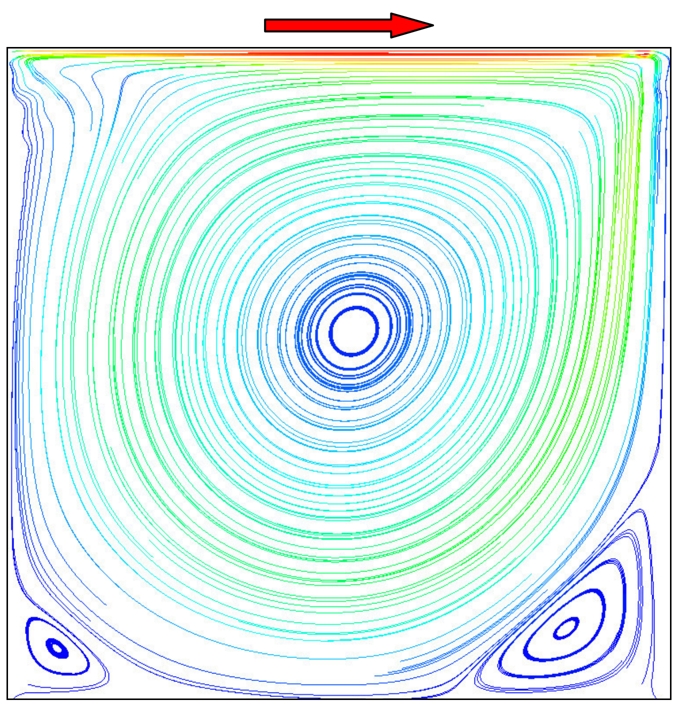


Figure C.7 – Streamlines for the Lattice Boltzmann LDCF1 Case

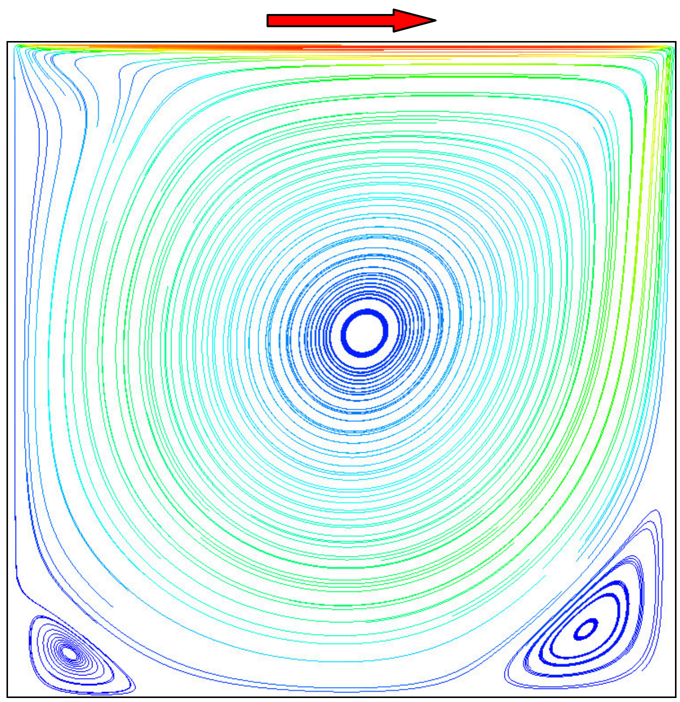


Figure C.8 – Streamlines for the Lattice Boltzmann LDCF4 Case

Velocity Centerline Plots for LDCF Cases

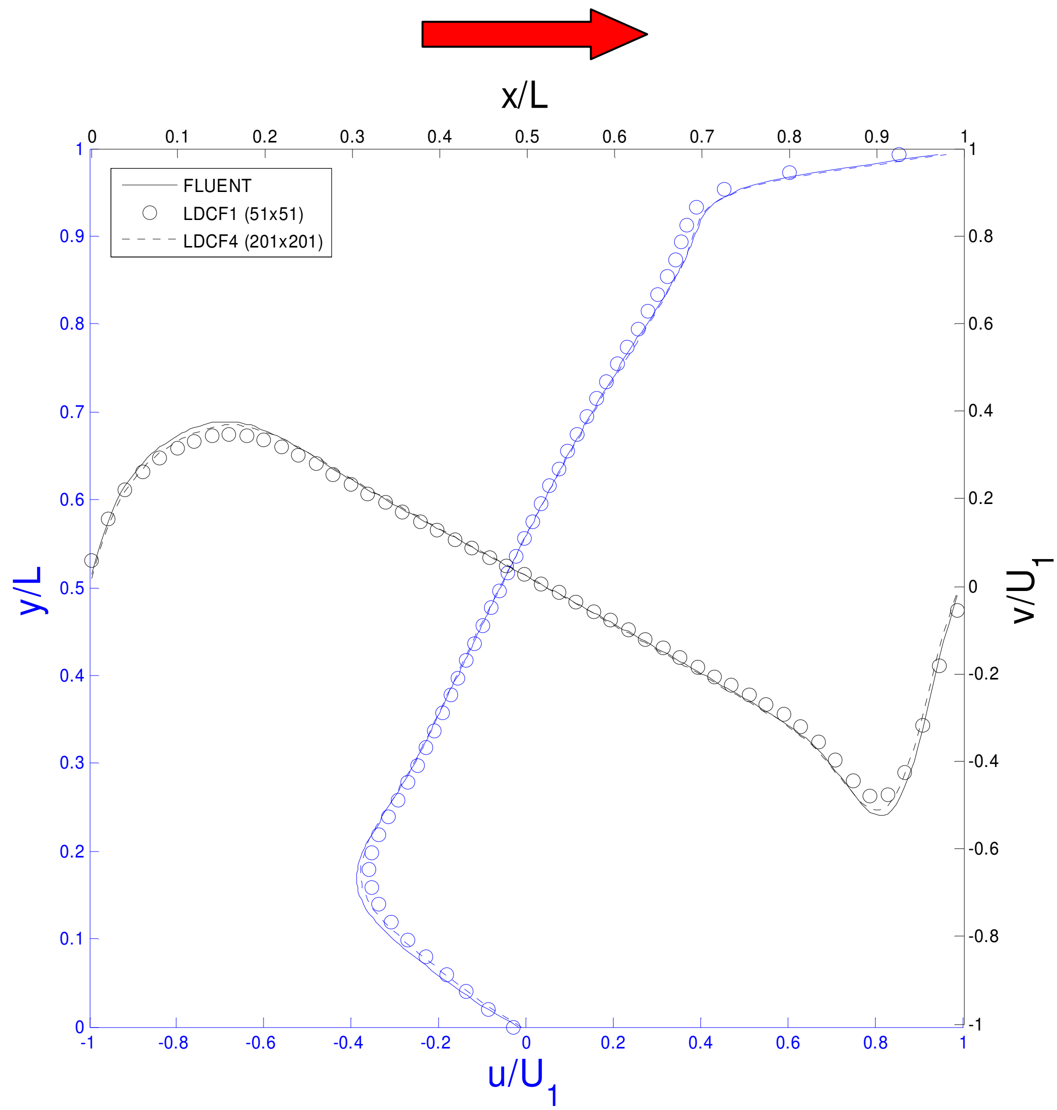


Figure C.9 – LDCF Velocity Plots using the SBB Treatment

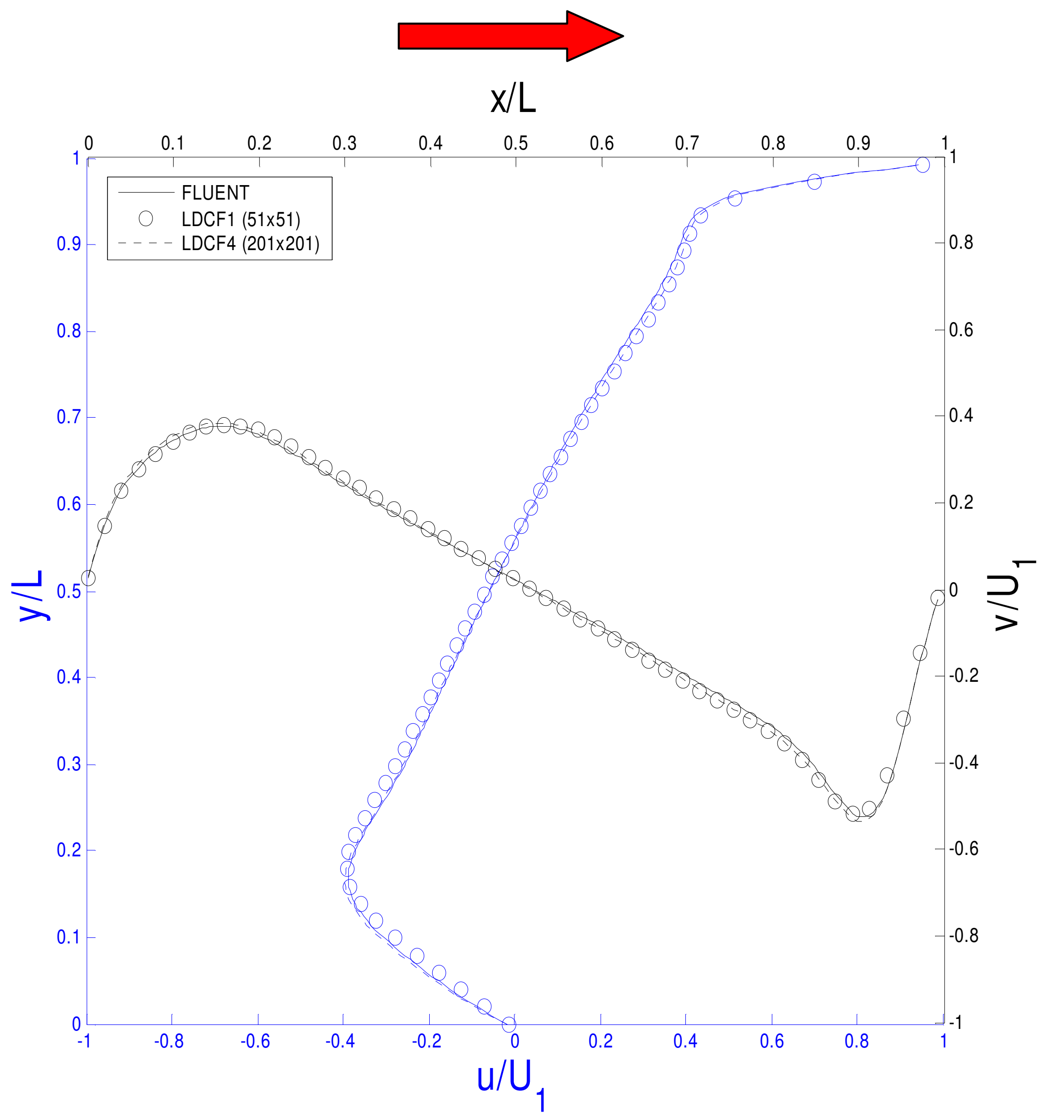


Figure C.10 – LDCF Velocity Plots using the MDI Treatment

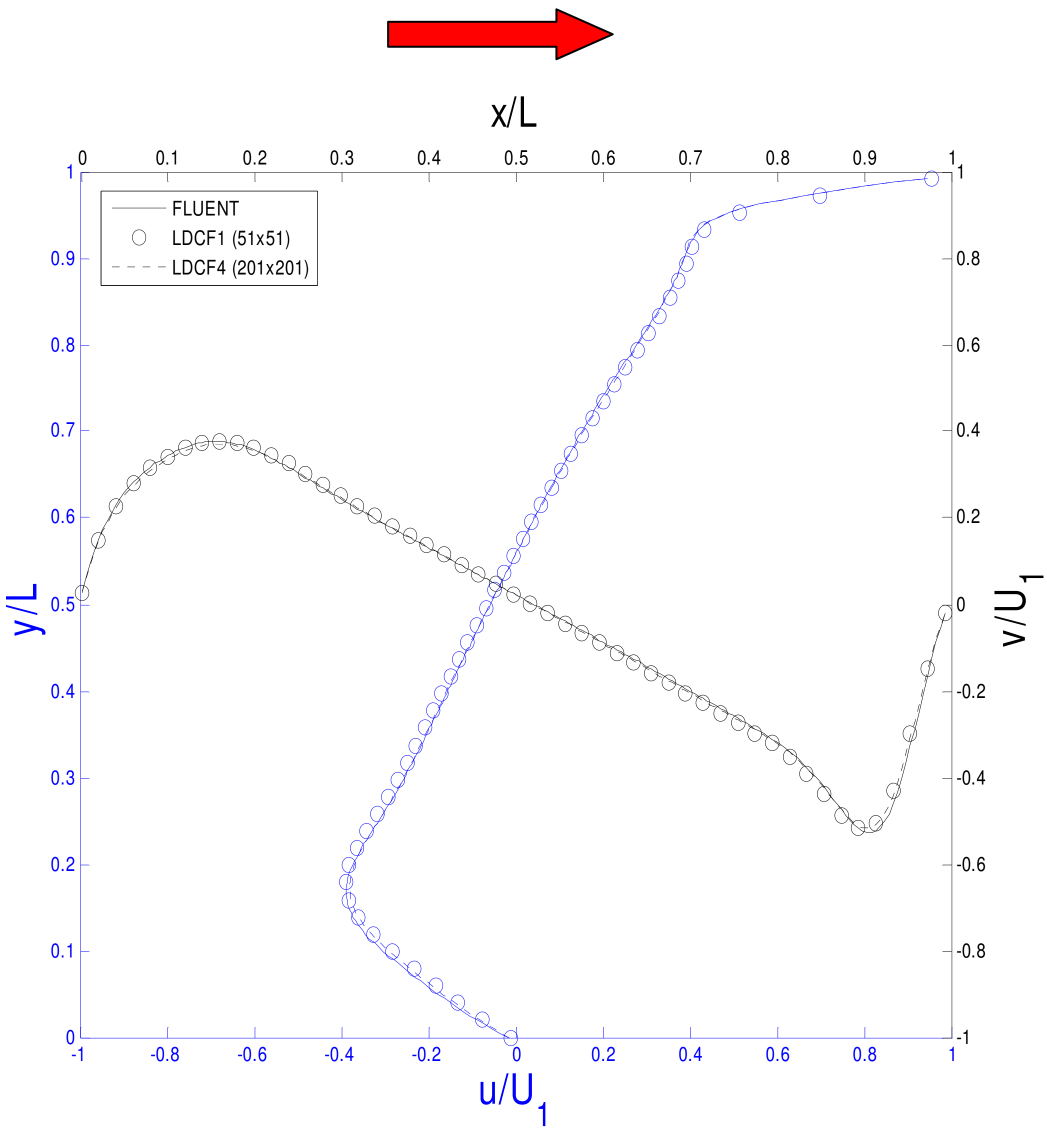


Figure C.11 – LDCF Velocity Plots using the BZ Treatment

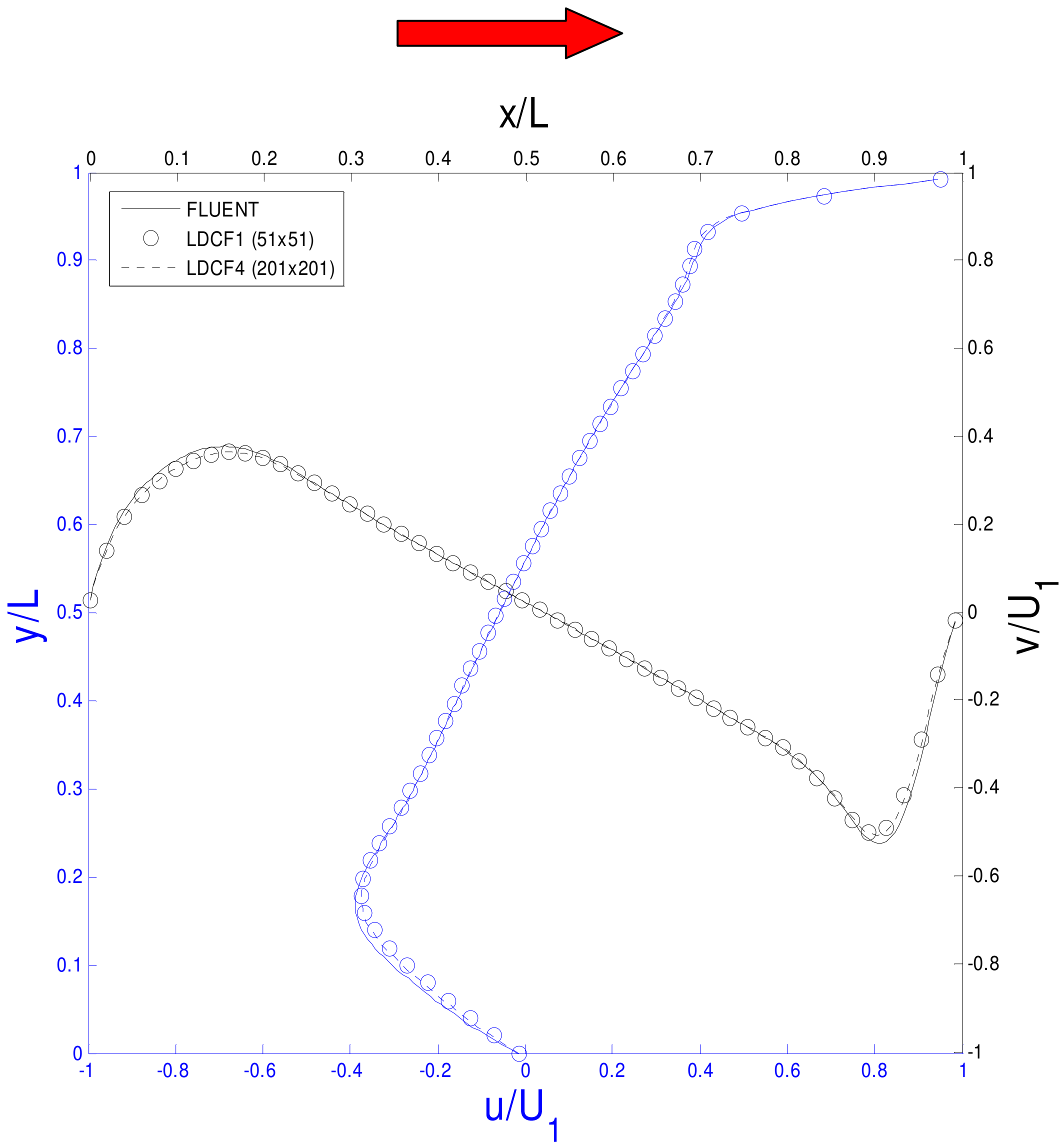


Figure C.12 – LDCF Velocity Plots using the YU Treatment

Absolute Error Visualizations for LDCF Cases

Note: LDCF bottom left corner is at origin

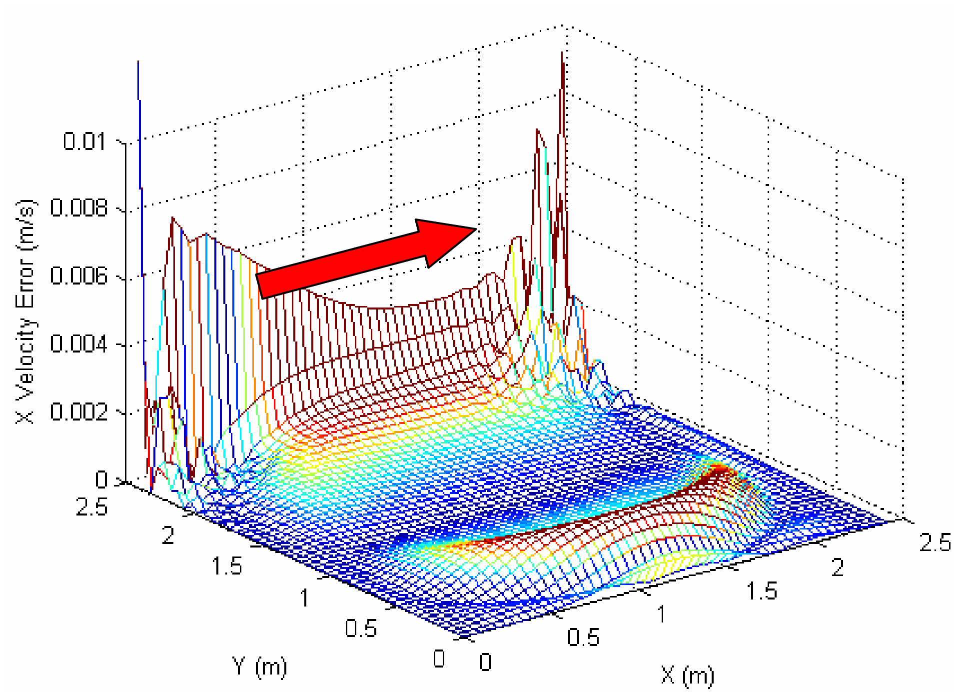


Figure C.13 – X-Velocity Error for the LDCF1 case using the SBB Treatment

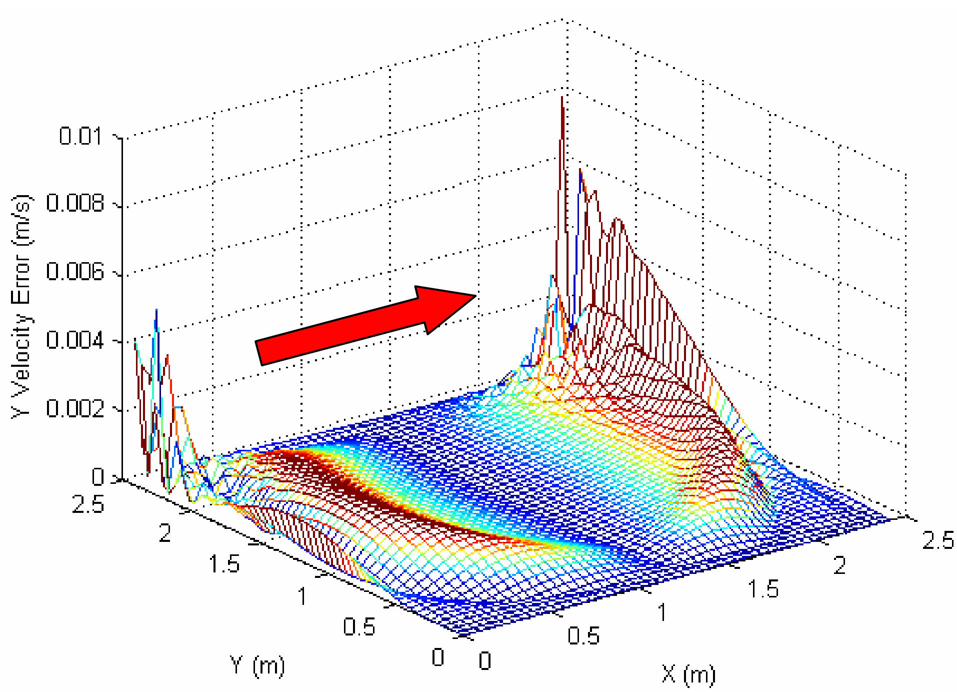


Figure C.14 – Y-Velocity Error for the LDCF1 case using SBB Treatment

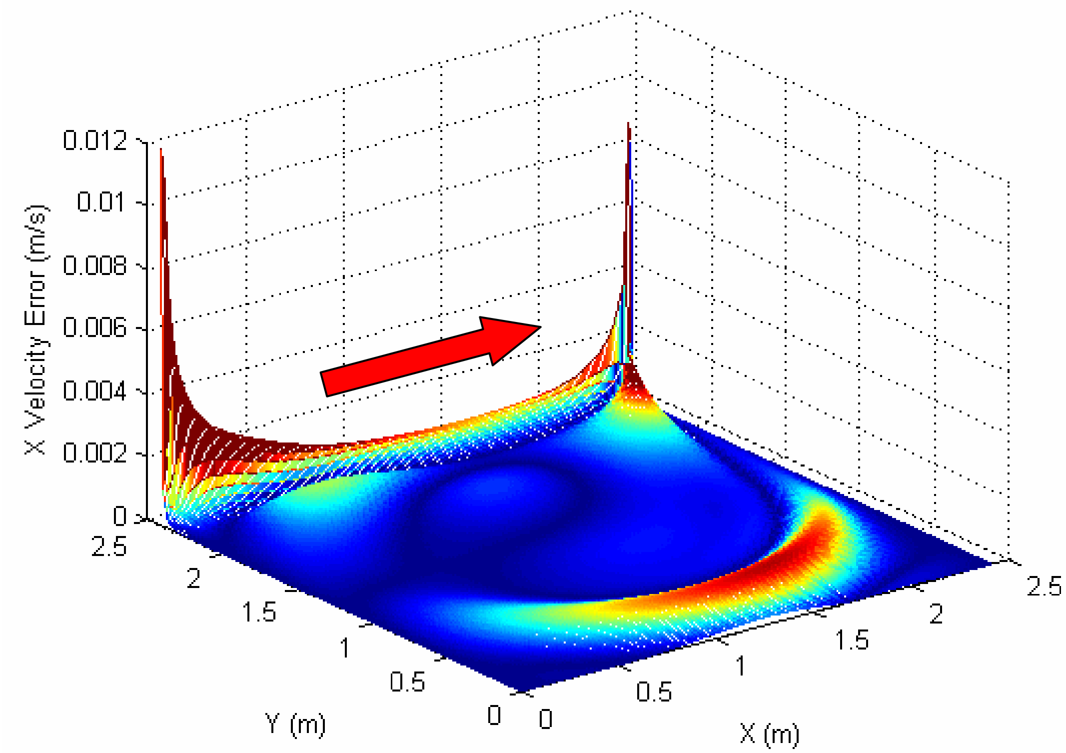


Figure C.15 – X-Velocity Error for the LDCF4 case using the SBB Treatment

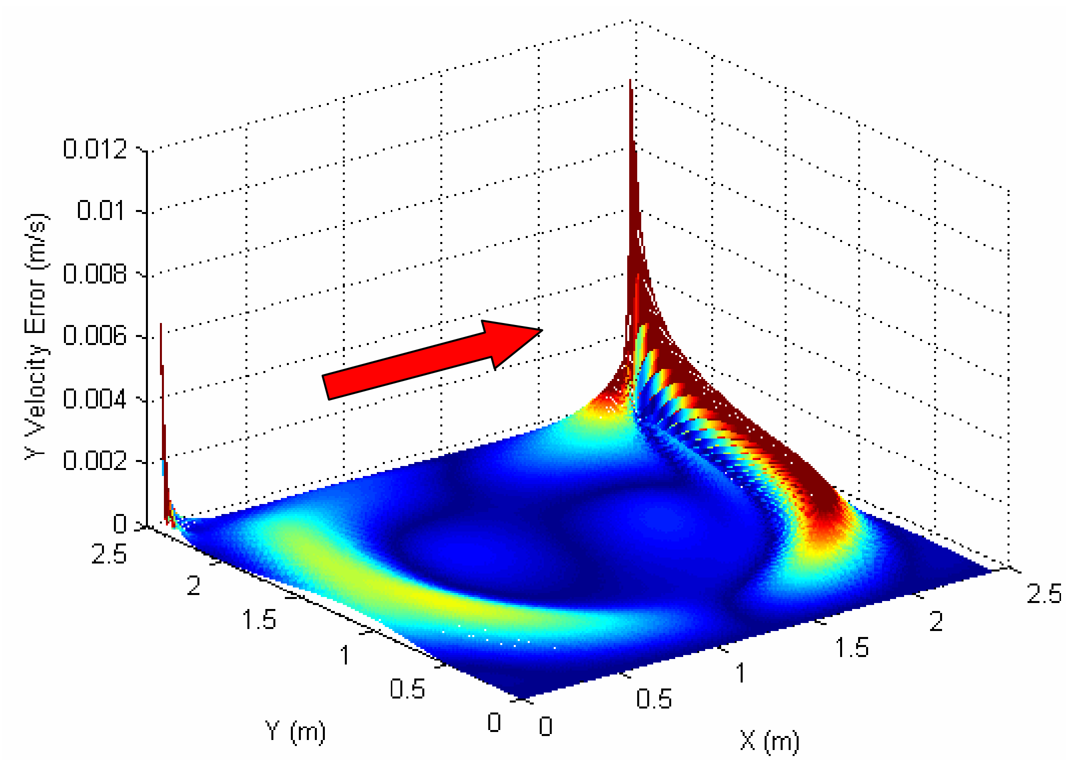


Figure C.16 – Y-Velocity Error for the LDCF4 case using the SBB Treatment

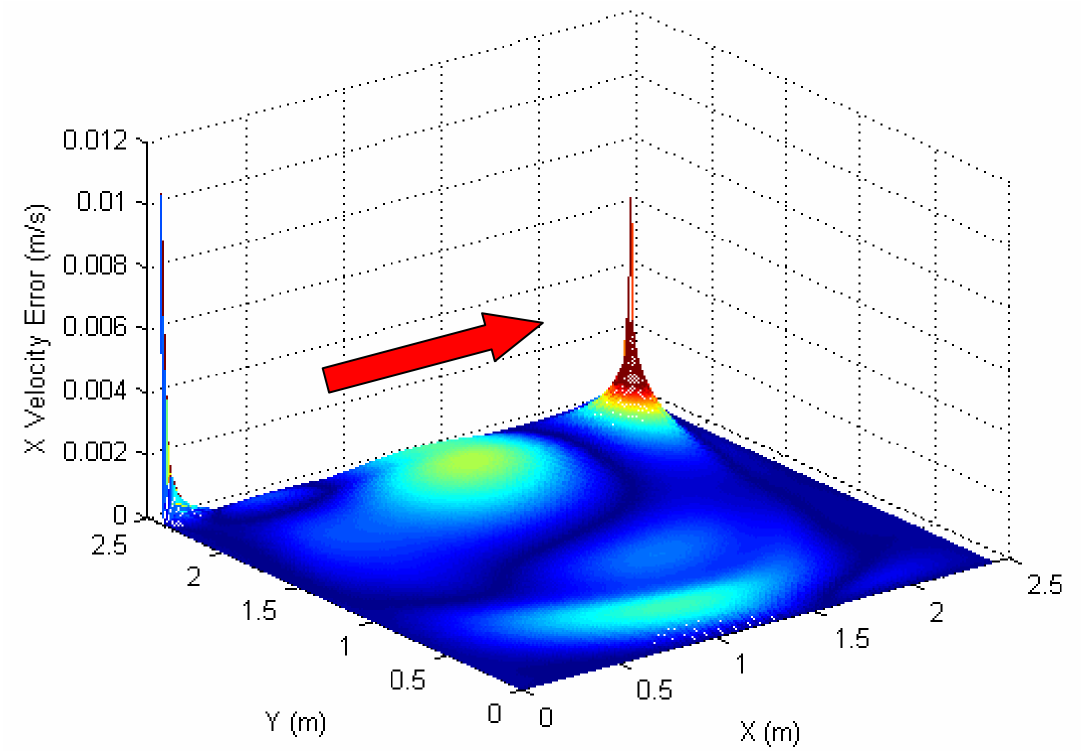


Figure C.17 – X-Velocity Error for the LDCF4 case using the MDI Treatment

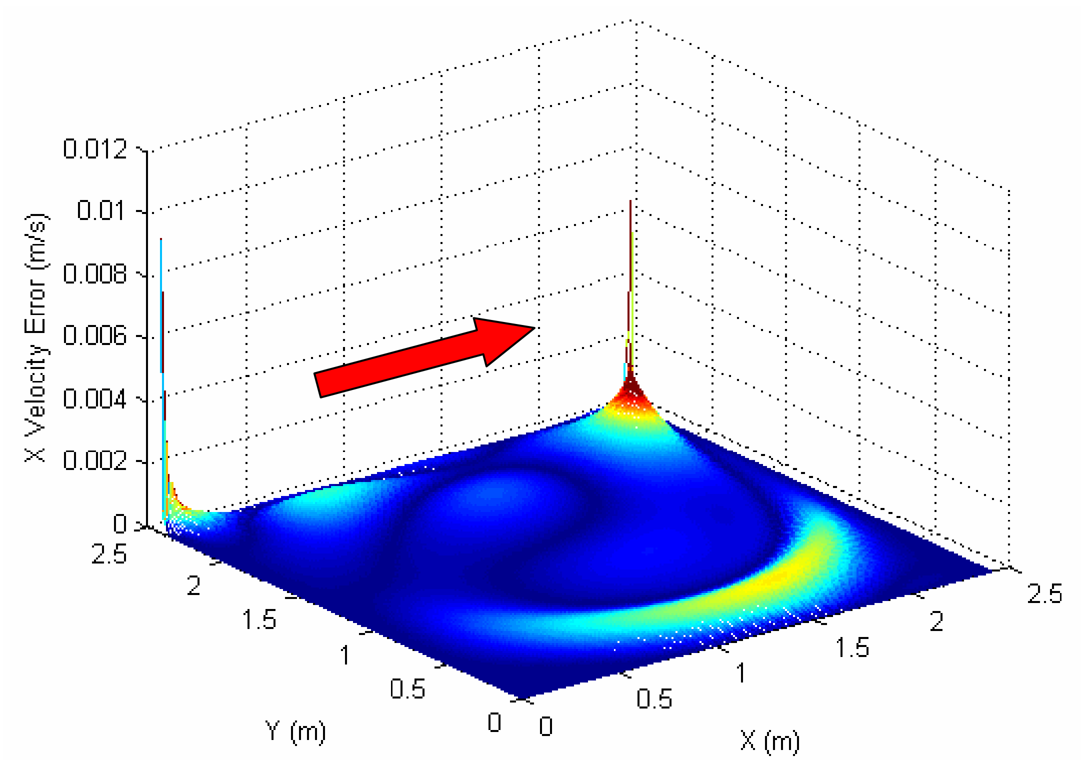


Figure C.18 – X-Velocity Error for the LDCF4 case using the BZ Treatment

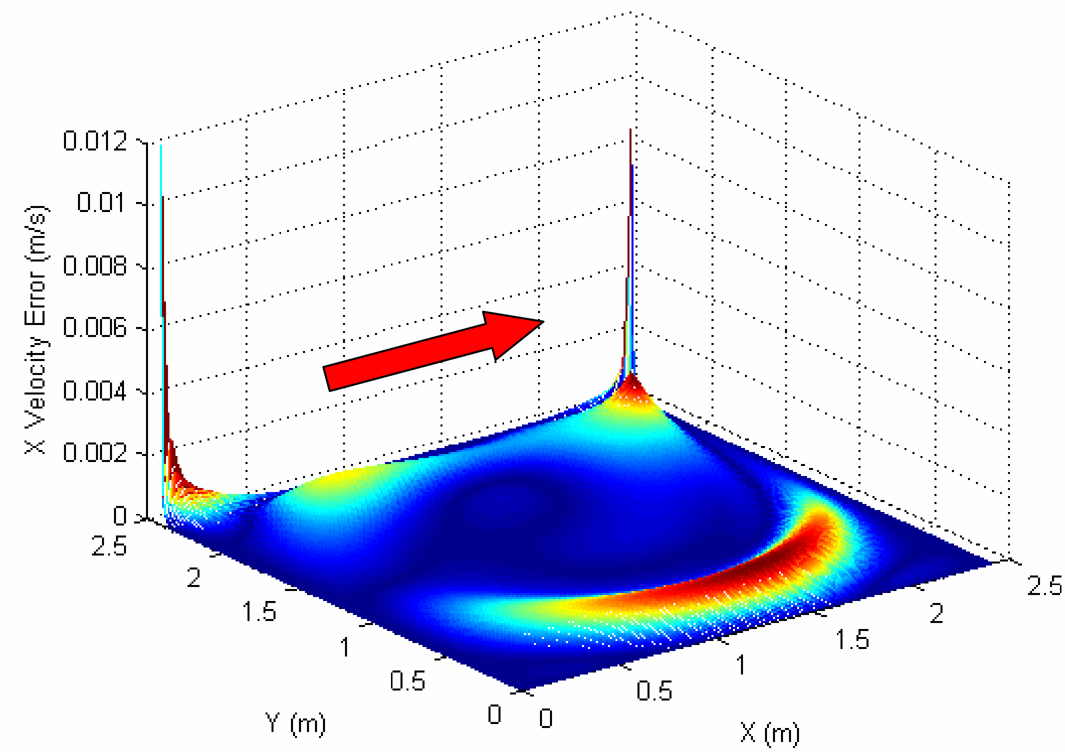


Figure C.19 – X-Velocity Error for the LDCF4 case using the YU Treatment

RMS Error Plots

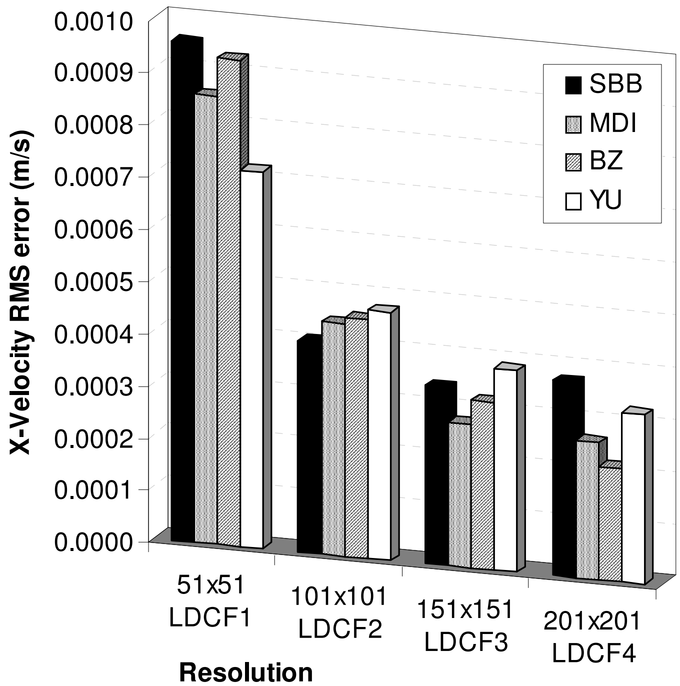


Figure C.20 – X-Velocity RMS Error for the LDCF Case

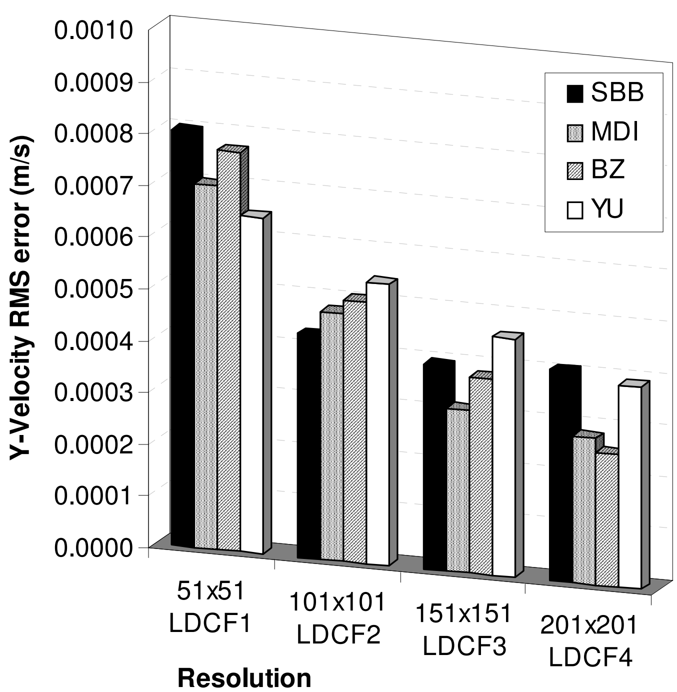


Figure C.21 – Y-Velocity RMS Error for the LDCF Case

Percent Error Results

Table C.2 – X-Velocity Percent Error Results for the LDCF Cases

Case	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1 51x51	SBB	4.45	31.71	71.07
	MDI	8.12	37.49	64.70
	BZ	8.98	47.13	74.19
	YU	13.75	47.09	68.68
LDCF2 101x101	SBB	16.10	59.75	87.28
	MDI	16.18	51.55	73.14
	BZ	18.76	50.39	77.43
	YU	15.30	47.73	72.95
LDCF3 151x151	SBB	24.05	57.89	83.36
	MDI	12.96	72.29	87.81
	BZ	25.25	59.89	86.82
	YU	20.21	52.48	81.86
LDCF4 201x201	SBB	17.96	57.85	78.32
	MDI	13.68	71.07	86.91
	BZ	23.35	76.38	89.87
	YU	24.29	58.42	85.73

Table C.3 – Y-Velocity Percent Error Results for the LDCF Cases

Case	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1 51x51	SBB	1.84	23.59	70.62
	MDI	9.94	42.29	71.36
	BZ	12.09	54.07	75.27
	YU	20.19	50.04	71.64
LDCF2 101x101	SBB	12.31	51.66	84.12
	MDI	18.22	49.50	75.21
	BZ	16.46	46.26	75.76
	YU	9.45	42.35	72.63
LDCF3 151x151	SBB	21.39	53.66	80.59
	MDI	12.29	73.40	87.39
	BZ	21.29	54.67	84.01
	YU	15.25	46.73	78.85
LDCF4 201x201	SBB	17.93	55.37	77.13
	MDI	14.11	71.25	85.76
	BZ	19.84	73.03	88.38
	YU	20.71	52.43	83.68

Flow in an Annulus

Absolute Error Visualization Plots for Annulus Cases

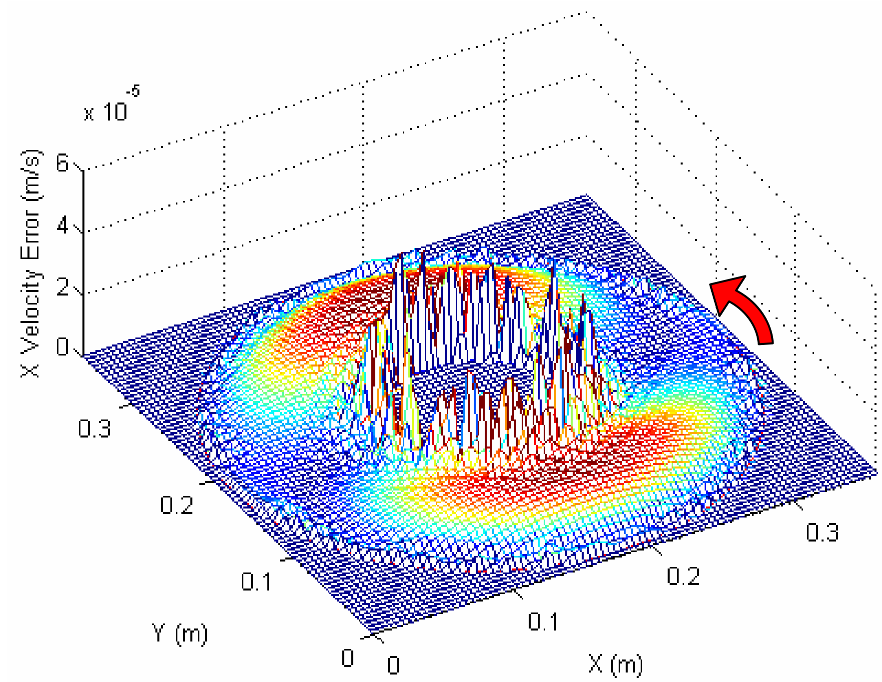


Figure C.22 – Typical X-Velocity Error for Taylor-Couette Case

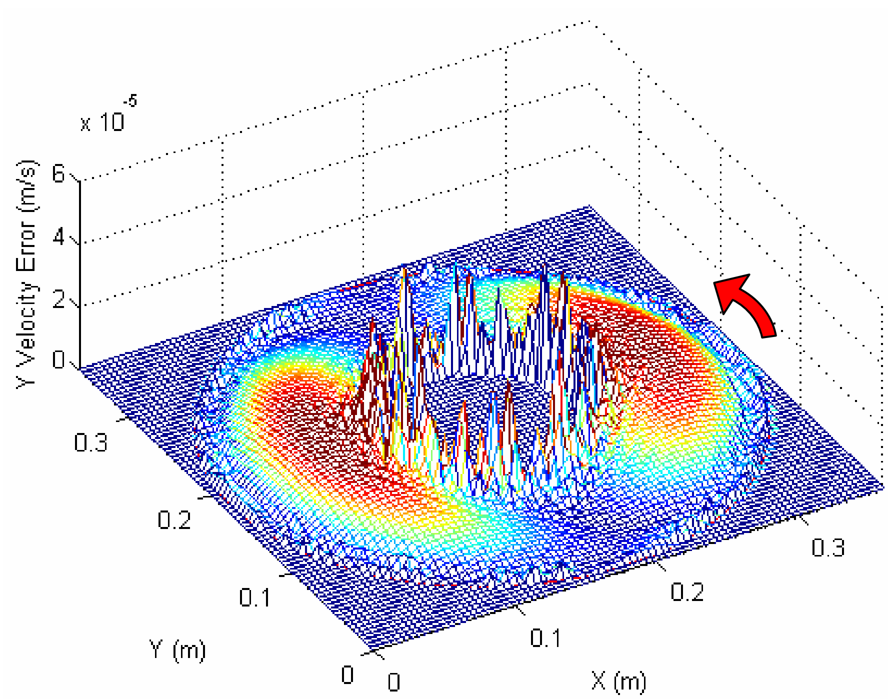


Figure C.23 – Typical Y-Velocity Error for Taylor-Couette Case

RMS Error Plots for Annulus Cases

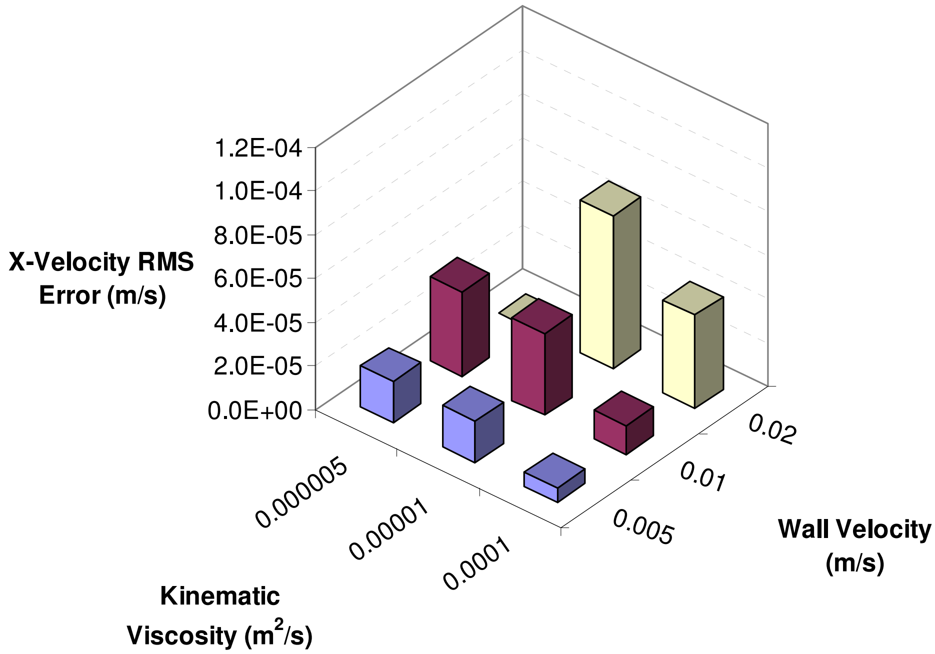


Figure C.24 – X-Velocity RMS Error for the ANN3 grid using the SBB Treatment

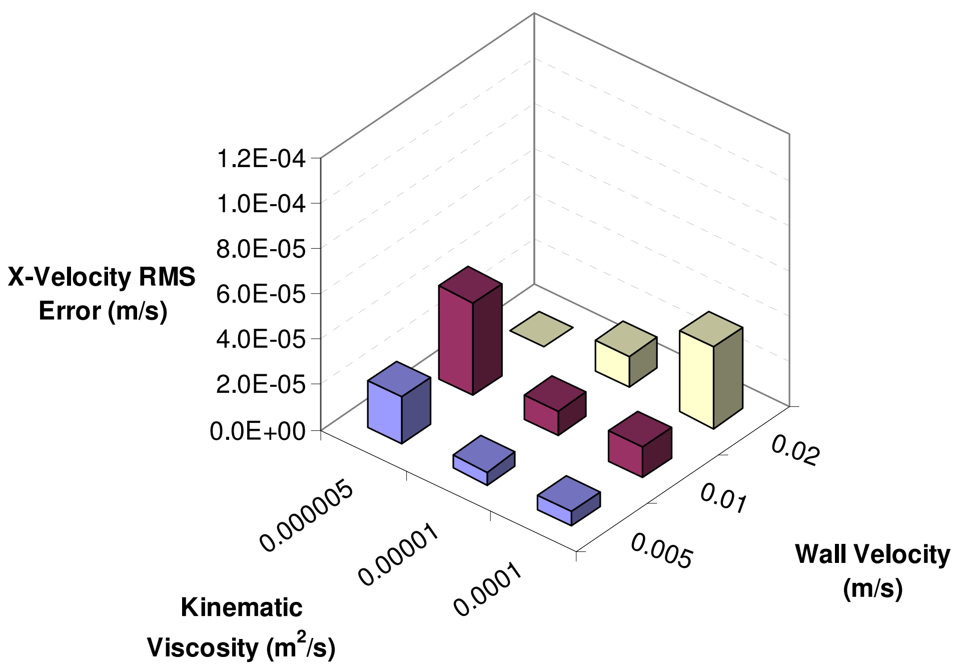


Figure C.25 – X-Velocity RMS Error for the ANN3 grid using the MDI Treatment

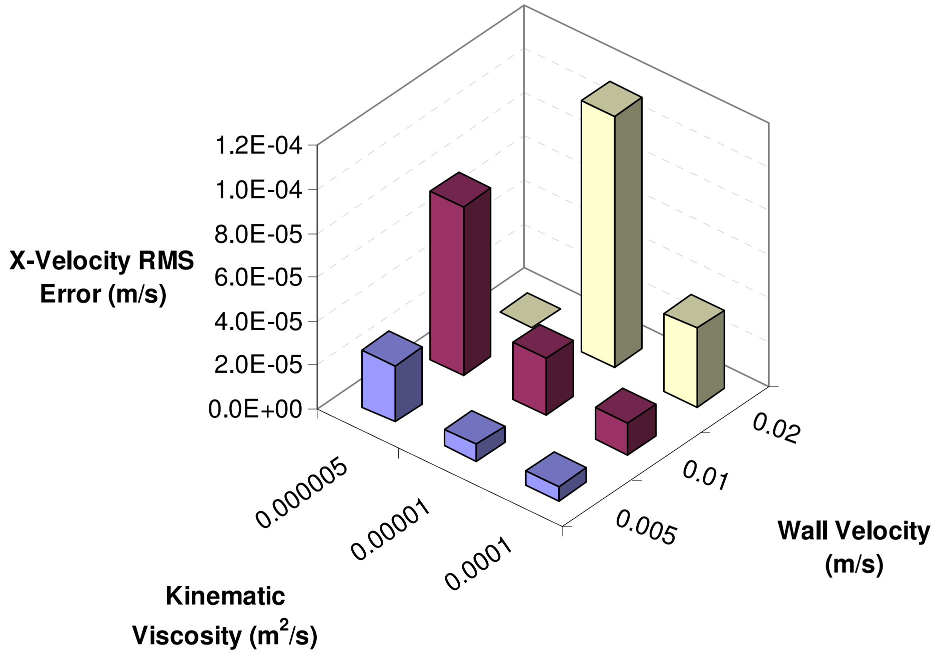


Figure C.26 – X-Velocity RMS Error for the ANN3 grid using the BZ Treatment

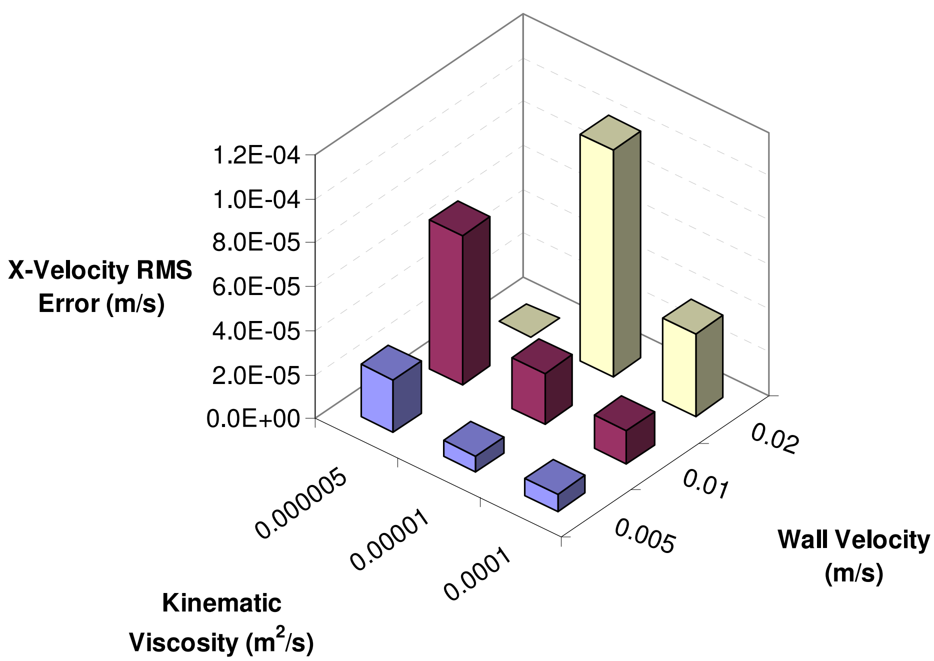


Figure C.27 – X-Velocity RMS Error for the ANN3 grid using the YU Treatment

Percent Error Results for Annulus Cases

Table C.4 – X-Velocity Percent Error Results for the ANN-CAS5 Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-CAS5 41x41	SBB	50.39	94.77	99.23
	MDI	96.32	99.81	99.81
	BZ	97.38	99.81	99.81
	YU	94.28	99.81	99.81
ANN2-CAS5 81x81	SBB	75.36	99.71	99.95
	MDI	99.37	99.95	99.95
	BZ	98.16	99.95	99.95
	YU	98.25	99.95	99.95
ANN3-CAS5 121x121	SBB	89.81	99.96	99.98
	MDI	99.88	99.98	99.98
	BZ	98.43	99.98	99.98
	YU	99.07	99.98	99.98

Table C.5 – X-Velocity Percent Error Results for the ANN-CAS6 Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-CAS6 41x41	SBB	N/A	N/A	N/A
	MDI	N/A	N/A	N/A
	BZ	N/A	N/A	N/A
	YU	N/A	N/A	N/A
ANN2-CAS6 81x81	SBB	85.86	99.76	99.95
	MDI	98.72	99.95	99.95
	BZ	N/A	N/A	N/A
	YU	33.22	99.90	99.95
ANN3-CAS6 121x121	SBB	91.57	99.91	99.96
	MDI	99.19	99.96	99.96
	BZ	41.32	99.96	99.98
	YU	47.30	99.96	99.98

Table C.6 – X-Velocity Percent Error Results for the ANN-CAS8 Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-CAS8 41x41	SBB	N/A	N/A	N/A
	MDI	N/A	N/A	N/A
	BZ	N/A	N/A	N/A
	YU	N/A	N/A	N/A
ANN2-CAS8 81x81	SBB	78.25	99.86	99.95
	MDI	98.28	99.95	99.95
	BZ	61.15	99.86	99.95
	YU	49.61	99.95	99.95
ANN3-CAS8 121x121	SBB	91.99	99.94	99.96
	MDI	99.30	99.96	99.96
	BZ	59.10	99.94	99.96
	YU	66.53	99.94	99.96

Streamline Visualization of Annulus Construct Case

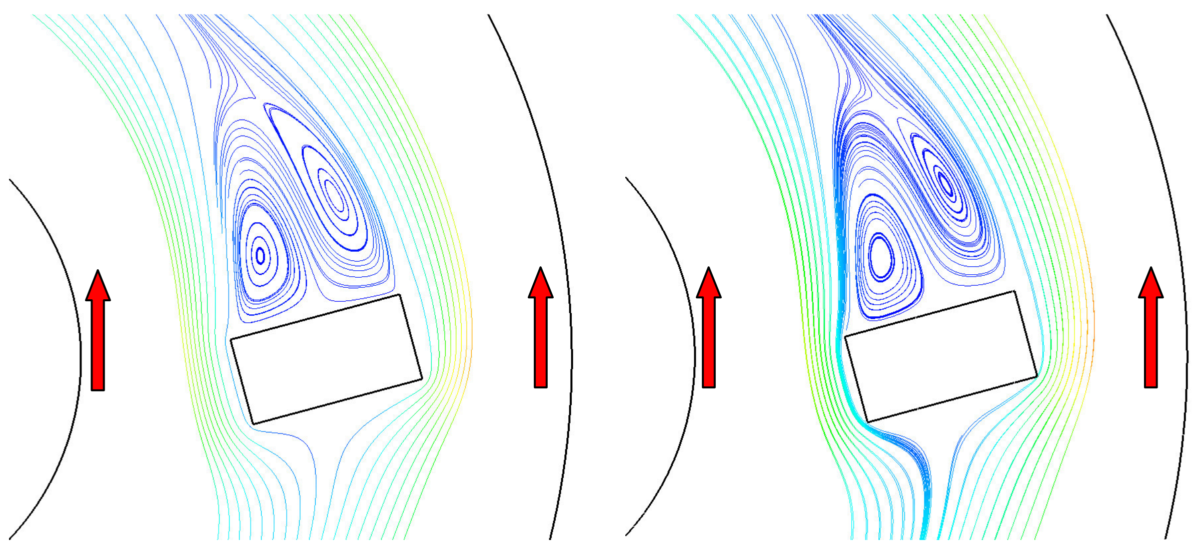


Figure C.28 – Streamline Comparison of FLUENT and LB Solutions for the ANN-RECT Case

Absolute Error Visualization Plots for Annulus Construct Cases

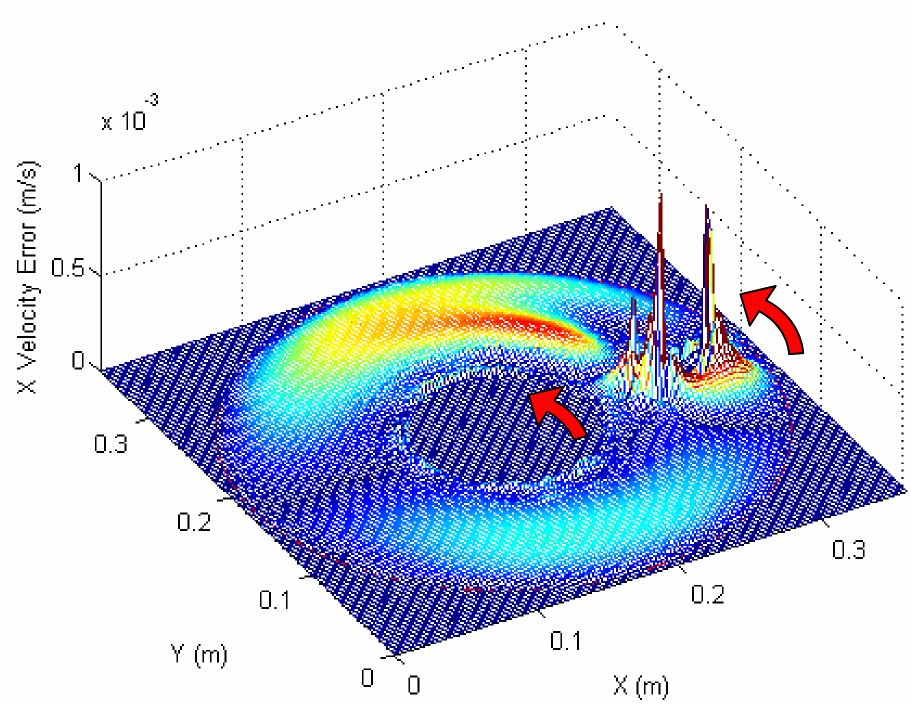


Figure C.29 – Typical X-Velocity Error for Annulus Construct Case

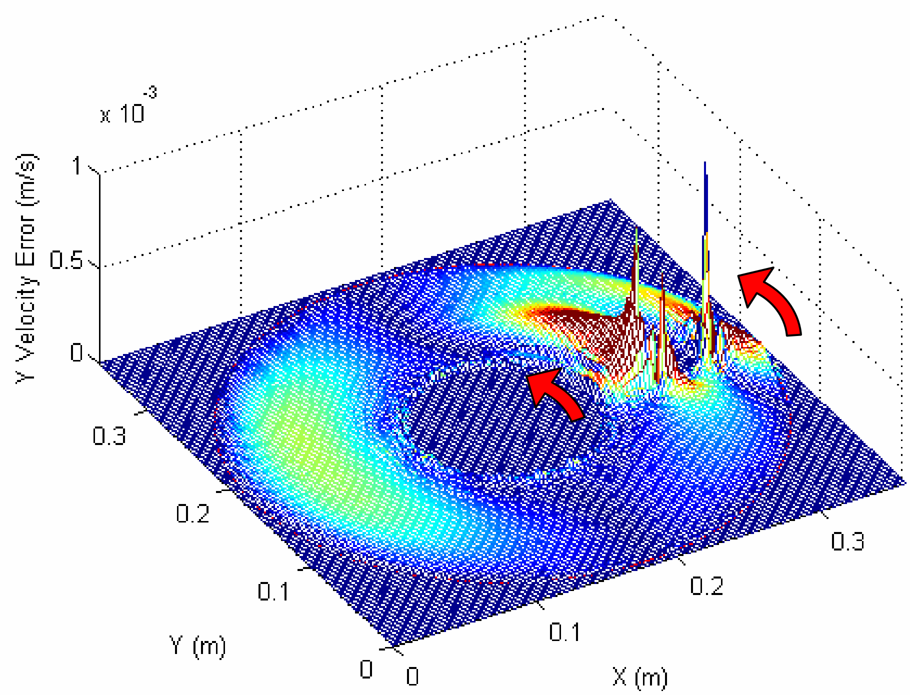


Figure C.30 – Typical Y-Velocity Error for Annulus Construct Case

RMS Error Plots for Annulus Construct Cases

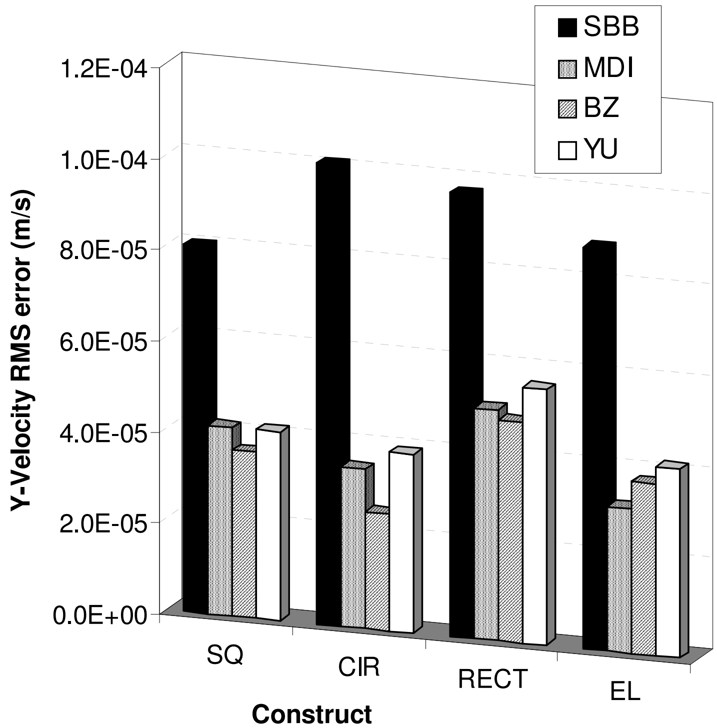


Figure C.31 – Y-Velocity RMS Error versus Construct for ANN3-CAS5 Cases

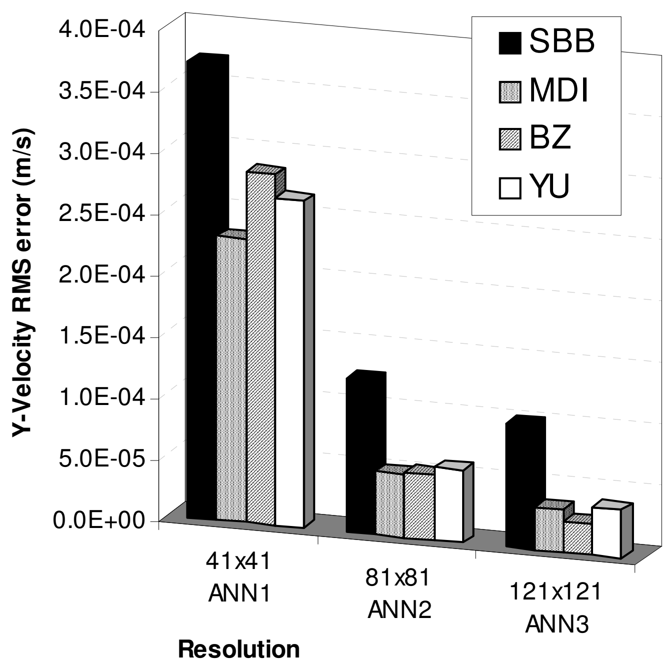


Figure C.32 – Y-Velocity RMS Error versus Resolution for ANN-CAS5-CIR Cases

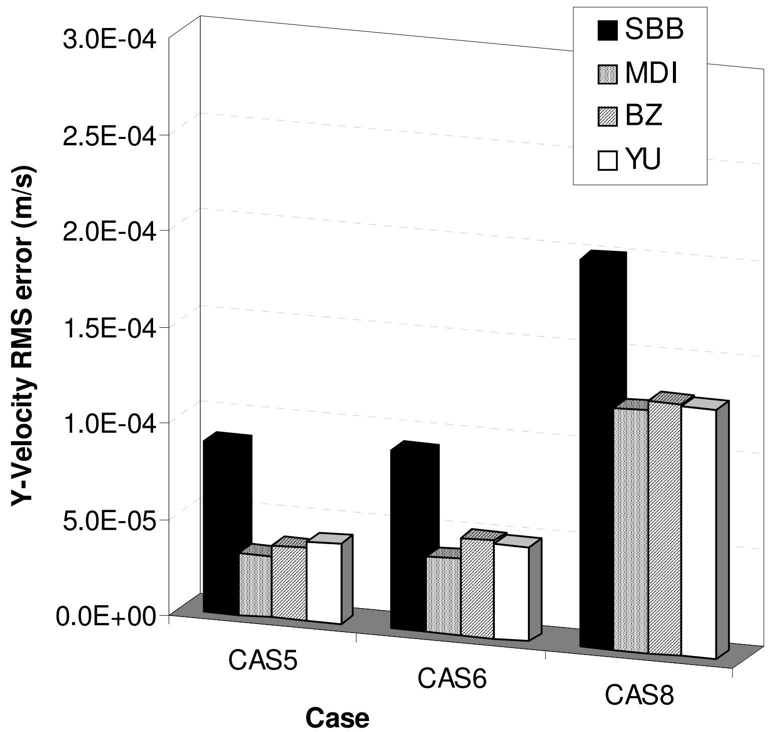


Figure C.33 Y-Velocity RMS Error versus Flow Parameters for ANN3-EL Cases

Percent Error Results for Annulus Construct Cases

Table C.7 – Y-Velocity Percent Error Results for the ANN3-CAS5 Construct Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN3-CAS5 SQ	SBB	33.60	88.87	96.86
	MDI	50.21	97.26	98.55
	BZ	49.30	96.55	98.15
	YU	47.15	96.39	98.12
ANN3-CAS5 CIR	SBB	29.30	76.60	96.64
	MDI	55.25	98.64	99.24
	BZ	62.71	98.09	99.06
	YU	48.93	97.54	98.80
ANN3-CAS5 RECT	SBB	39.89	90.17	95.32
	MDI	56.72	94.68	97.12
	BZ	60.86	94.77	97.05
	YU	51.64	93.88	96.74
ANN3-CAS5 EL	SBB	43.59	91.73	96.18
	MDI	68.74	97.86	98.90
	BZ	58.65	97.46	98.75
	YU	58.91	97.45	98.63

Table C.8 – Y-Velocity Percent Error Results for the ANN-CIR-CAS5 Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-CAS5 CIR	SBB	10.71	31.95	52.22
	MDI	16.89	46.43	71.62
	BZ	13.22	38.42	60.52
	YU	12.65	39.48	64.96
ANN2-CAS5 CIR	SBB	27.94	70.06	95.56
	MDI	47.87	97.22	98.15
	BZ	40.28	96.42	98.00
	YU	40.67	95.86	98.10
ANN3-CAS5 CIR	SBB	29.30	76.60	96.64
	MDI	55.25	98.64	99.24
	BZ	62.71	98.09	99.06
	YU	48.93	97.54	98.80

Table C.9 – Y-Velocity Percent Error Results for the ANN3-EL Flow Parameter Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN3-CAS5 EL	SBB	43.59	91.73	96.18
	MDI	68.74	97.86	98.90
	BZ	58.65	97.46	98.75
	YU	58.91	97.45	98.63
ANN3-CAS6 EL	SBB	31.47	88.40	94.19
	MDI	59.90	95.64	98.16
	BZ	46.14	92.28	97.07
	YU	51.03	93.35	97.47
ANN3-CAS8 EL	SBB	30.51	75.45	94.52
	MDI	46.61	89.62	98.01
	BZ	39.07	81.66	97.15
	YU	39.63	84.38	97.39

Absolute Error Visualizations for Annulus Density Cases

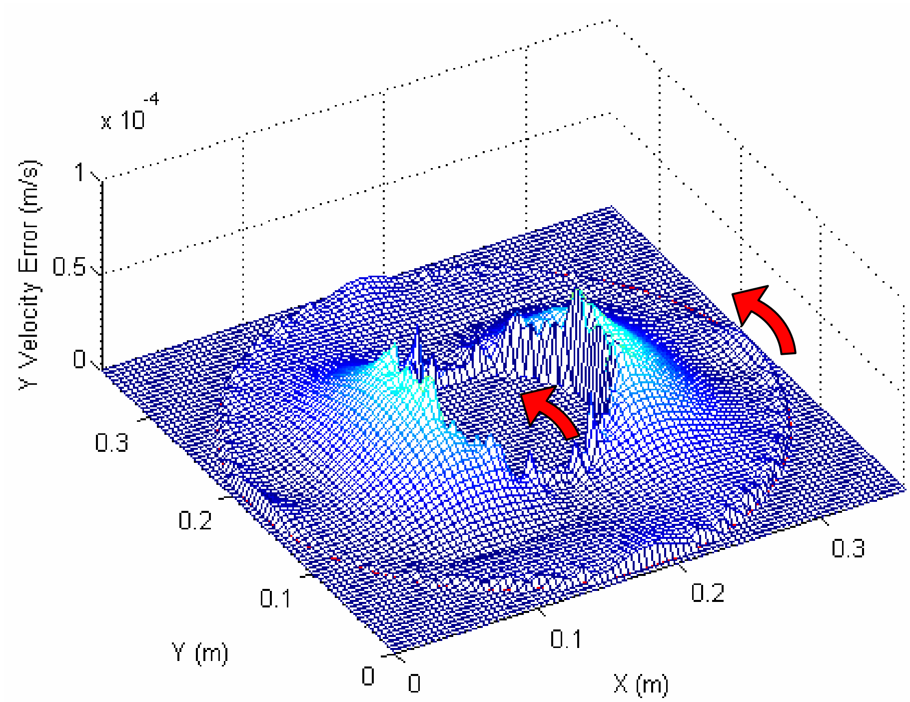


Figure C.34 – Y-Velocity Absolute Error for Annulus Local Density Case

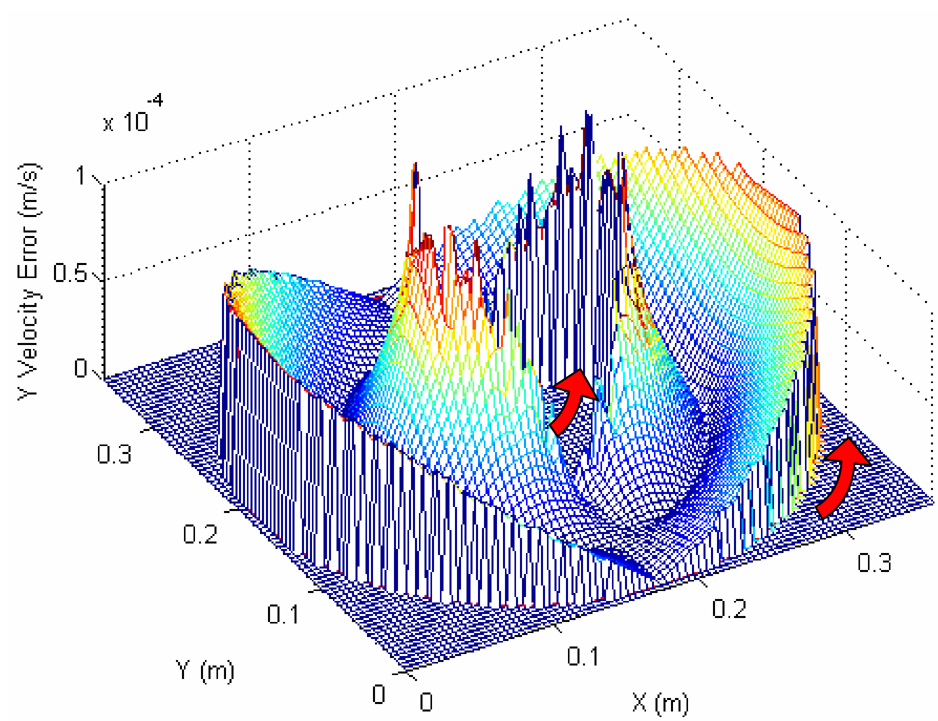


Figure C.35 – Y-Velocity Absolute Error for Annulus Average Density Case

RMS Error Results for Annulus Density Cases

Table C.10 – RMS Error Results for the CAS4-ANN2 Density Cases

Grid	Boundary Treatment	RMS error X-Velocity (m/s)	RMS error Y-Velocity (m/s)	RMS error V magnitude (m/s)
ANN2-CAS4 Local Density	SBB	0.000016	0.000016	0.000018
	MDI	0.000011	0.000010	0.000014
	BZ	0.000013	0.000013	0.000018
	YU	0.000015	0.000015	0.000021
ANN2-CAS4 Average Density	SBB	0.000057	0.000057	0.000079
	MDI	0.000041	0.000041	0.000057
	BZ	0.000040	0.000040	0.000056
	YU	0.000040	0.000040	0.000056

Absolute Error Visualization Plots for the Refined LDCF Cases

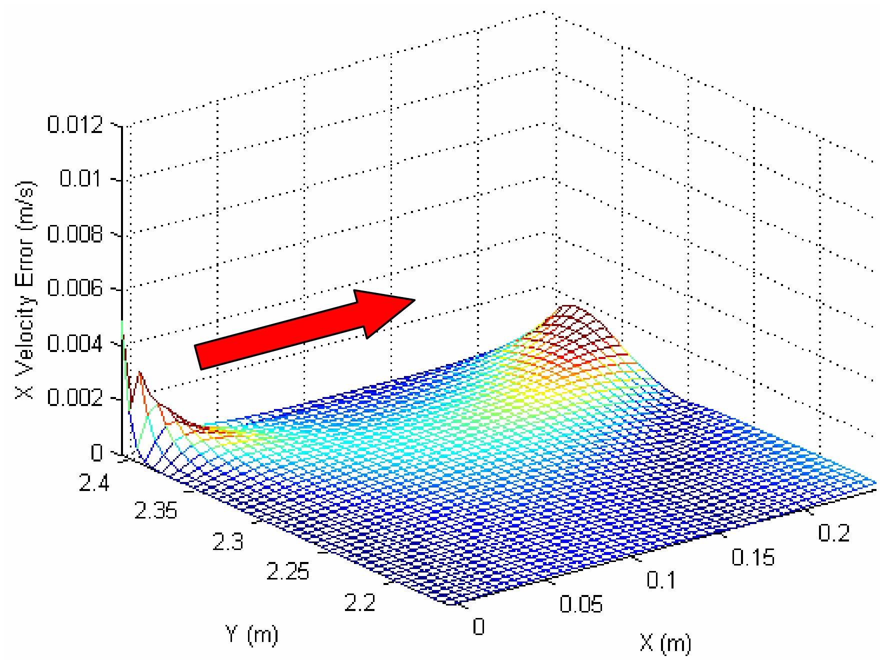


Figure C.36 – Absolute X-Velocity Error for the LDCF4-RF2-UL Case

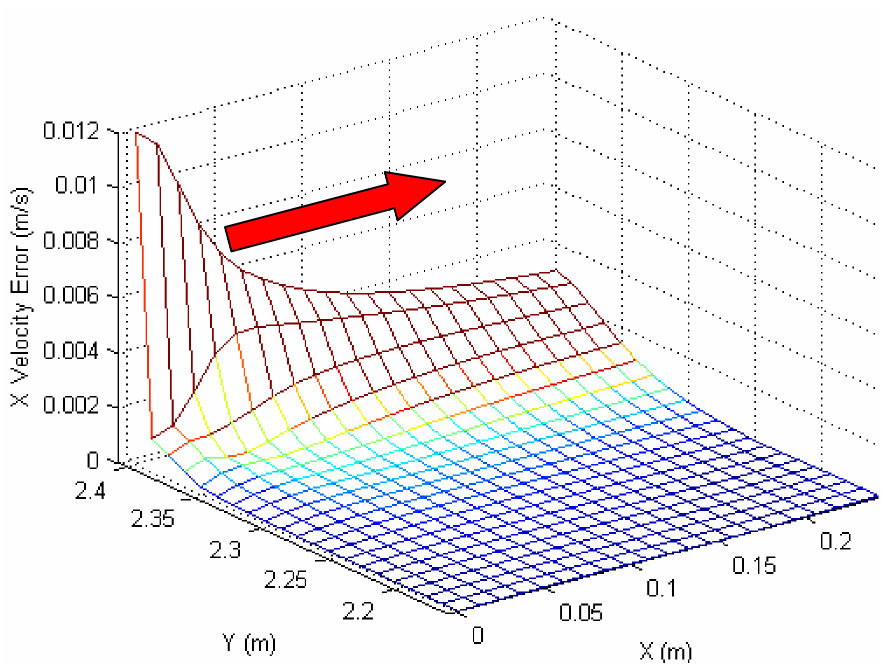


Figure C.37 – Absolute X-Velocity Error for the LDCF4 Case

RMS Error Plots for the Refined LDCF Cases

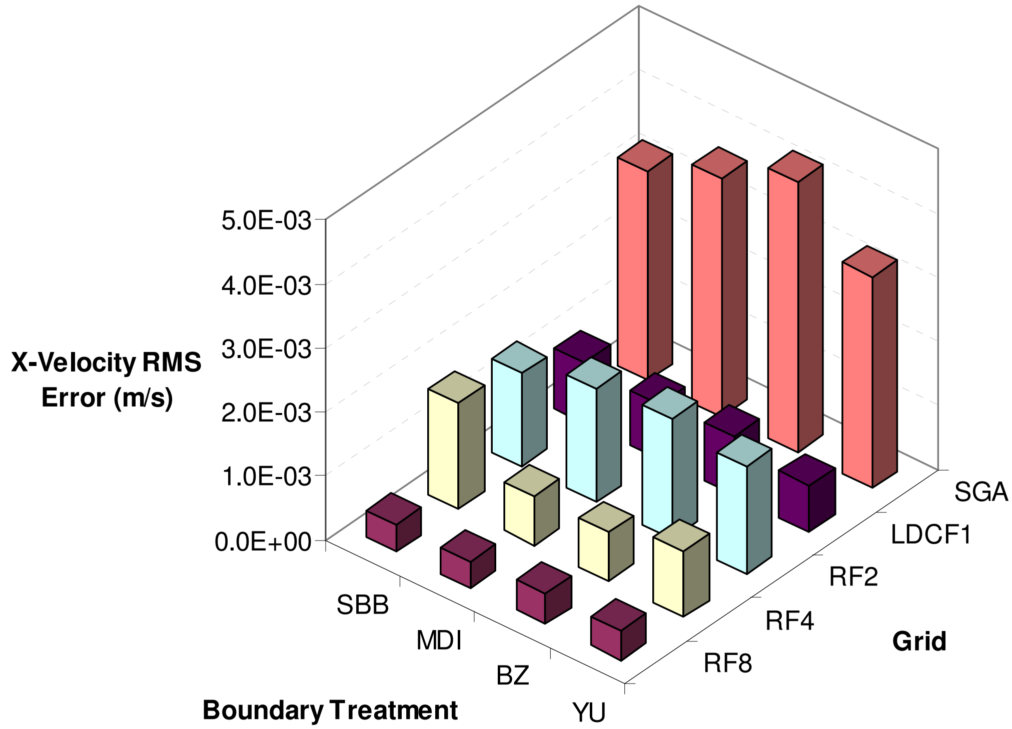


Figure C.38 – X-Velocity RMS Error for the LDCF1-UL Grid Refinement

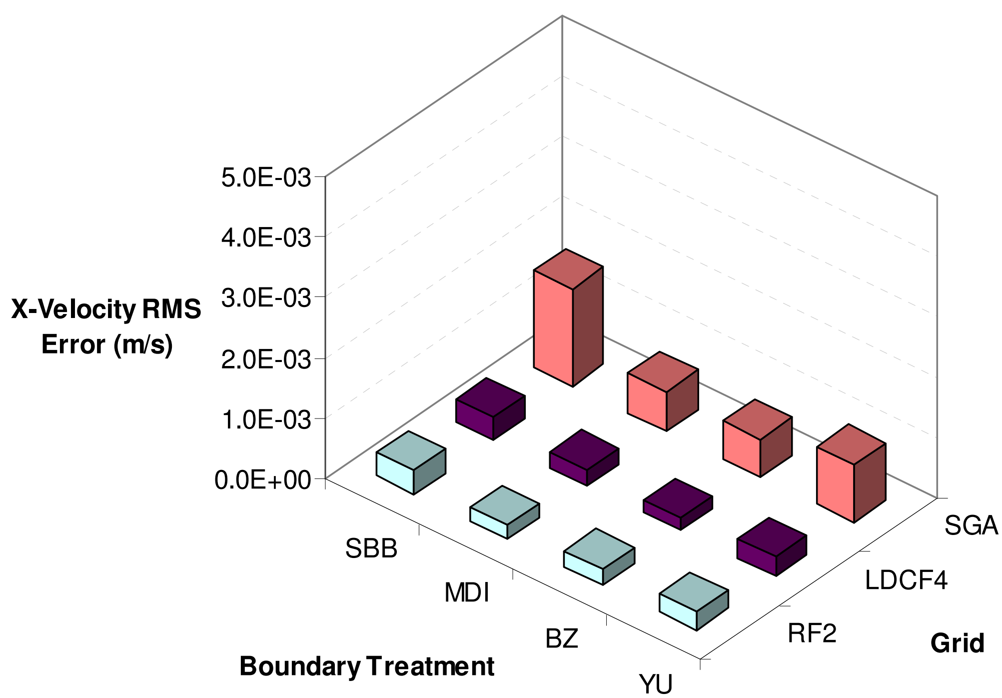


Figure C.39 – X-Velocity RMS Error for the LDCF4-UL Grid Refinement

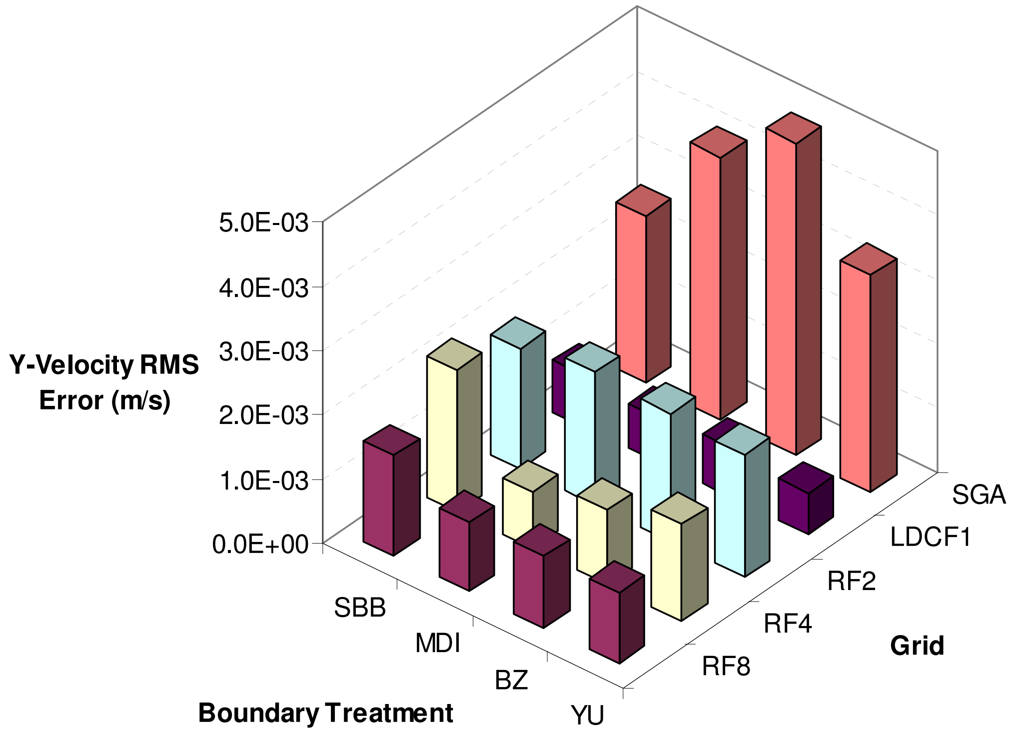


Figure C.40 – Y-Velocity RMS Error for the LDCF1-UR Grid Refinement

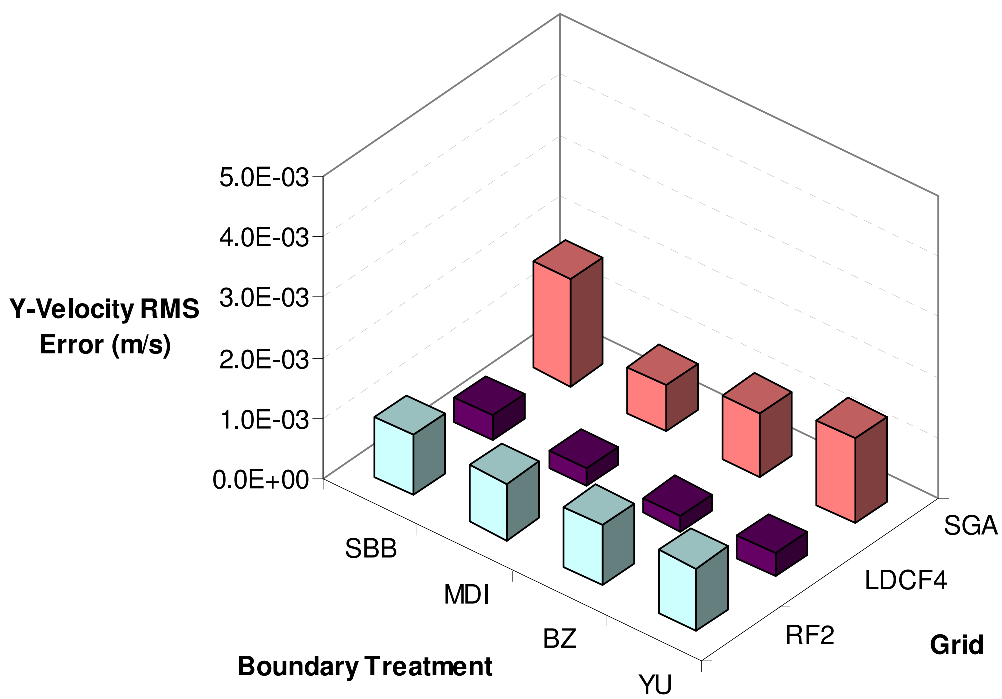


Figure C.41 – Y-Velocity RMS Error for the LDCF4-UR Grid Refinement

Absolute Error Results for the Refined LDCF Cases

Table C.11 – Percent Decrease in X-Velocity RMS Error for LDCF-UL Cases

Grid	Boundary Treatment	RMS error X-Velocity Sub-Grid (m/s)	RMS error X-Velocity SGA (m/s)	% Decrease in RMS error
LDCF1-UL RF8	SBB	0.000428	0.003276	86.93%
	MDI	0.000416	0.003722	88.83%
	BZ	0.000476	0.004233	88.75%
	YU	0.000463	0.003312	86.02%
LDCF2-UL RF4	SBB	0.000291	0.001411	79.37%
	MDI	0.000330	0.001733	80.93%
	BZ	0.000361	0.001856	80.56%
	YU	0.000341	0.001667	79.55%
LDCF4-UL RF2	SBB	0.000389	0.001614	75.92%
	MDI	0.000245	0.000666	63.23%
	BZ	0.000266	0.000625	57.51%
	YU	0.000302	0.000959	68.56%

Table C.12 – Percent Decrease in Y-Velocity RMS Error for LDCF-UR Cases

Grid	Boundary Treatment	RMS error Y-Velocity Sub-Grid (m/s)	RMS error Y-Velocity SGA (m/s)	% Decrease in RMS error
LDCF1-UR RF8	SBB	0.001552	0.002607	40.50%
	MDI	0.001074	0.004079	73.67%
	BZ	0.001113	0.004866	77.14%
	YU	0.001108	0.003368	67.12%
LDCF2-UR RF4	SBB	0.001124	0.001611	30.24%
	MDI	0.001045	0.001898	44.96%
	BZ	0.001068	0.002005	46.73%
	YU	0.001082	0.001870	42.17%
LDCF4-UR RF2	SBB	0.000979	0.001763	44.44%
	MDI	0.000933	0.000776	-20.19%
	BZ	0.001004	0.001059	5.24%
	YU	0.001038	0.001403	25.98%

Percent Error Results for the Refined LDCF Cases

Table C.13 – X-Velocity Percent Error Results for the LDCF-UL Sub-Grid Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1-UL RF8	SBB	5.76	13.00	15.74
	MDI	9.81	23.09	64.20
	BZ	10.49	23.55	67.85
	YU	8.49	21.49	48.97
LDCF2-UL RF4	SBB	8.44	32.95	79.19
	MDI	8.78	28.62	76.23
	BZ	8.55	26.68	69.44
	YU	8.67	27.48	70.35
LDCF4-UL RF2	SBB	8.21	26.34	67.67
	MDI	15.05	67.16	89.85
	BZ	13.46	47.21	83.64
	YU	9.41	34.27	75.83

Table C.14 – X-Velocity Percent Error Results for the LDCF-UL SGA Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1-UL SGA	SBB	2.78	13.89	19.44
	MDI	0.00	5.56	11.11
	BZ	0.00	0.00	11.11
	YU	2.78	16.67	19.44
LDCF2-UL SGA	SBB	5.79	33.88	64.46
	MDI	6.61	45.46	61.16
	BZ	6.61	43.80	63.64
	YU	0.00	14.88	56.20
LDCF4-UL SGA	SBB	5.22	29.71	50.34
	MDI	41.50	79.82	87.53
	BZ	4.08	29.71	76.42
	YU	0.68	9.52	59.18

Table C.15 – Y-Velocity Percent Error Results for the LDCF-UR Sub-Grid Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1-UR RF8	SBB	0.00	0.06	28.52
	MDI	6.35	28.57	53.29
	BZ	5.84	27.95	51.64
	YU	1.76	22.90	49.43
LDCF2-UR RF4	SBB	0.00	18.08	44.73
	MDI	3.34	22.73	47.96
	BZ	1.53	22.11	47.39
	YU	0.00	20.98	47.00
LDCF4-UR RF2	SBB	10.54	28.12	50.79
	MDI	9.18	30.61	51.30
	BZ	4.42	25.45	48.36
	YU	1.98	22.73	47.85

Table C.16 – Y-Velocity Percent Error Results for the LDCF-UR SGA Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1-UR SGA	SBB	0.00	5.56	25.00
	MDI	0.00	16.67	27.78
	BZ	5.56	19.44	30.56
	YU	2.78	22.22	38.89
LDCF2-UR SGA	SBB	1.65	27.27	48.76
	MDI	13.22	33.88	49.59
	BZ	11.57	33.88	52.07
	YU	19.01	39.67	57.85
LDCF4-UR SGA	SBB	6.58	31.97	47.62
	MDI	9.07	23.36	36.51
	BZ	13.61	28.57	46.26
	YU	15.87	35.37	53.06

RMS Error Results for the LDCF-LM Cases

Table C.17 – RMS X-Velocity Error Results for LDCF-LM Cases

Grid	Boundary Treatment	RMS error X-Velocity Sub-Grid (m/s)	RMS error X-Velocity Main Grid (m/s)
LDCF1-LM RF8	SBB	0.001032	0.001028
	MDI	0.000308	0.000306
	BZ	0.000428	0.000435
	YU	0.000562	0.000587
LDCF2-LM RF4	SBB	0.000567	0.000578
	MDI	0.000525	0.000538
	BZ	0.000588	0.000603
	YU	0.000678	0.000694
LDCF4-LM RF2	SBB	0.000332	0.000345
	MDI	0.000320	0.000294
	BZ	0.000228	0.000240
	YU	0.000456	0.000470

Absolute Error Visualization Plots for the Refined Annulus Cases

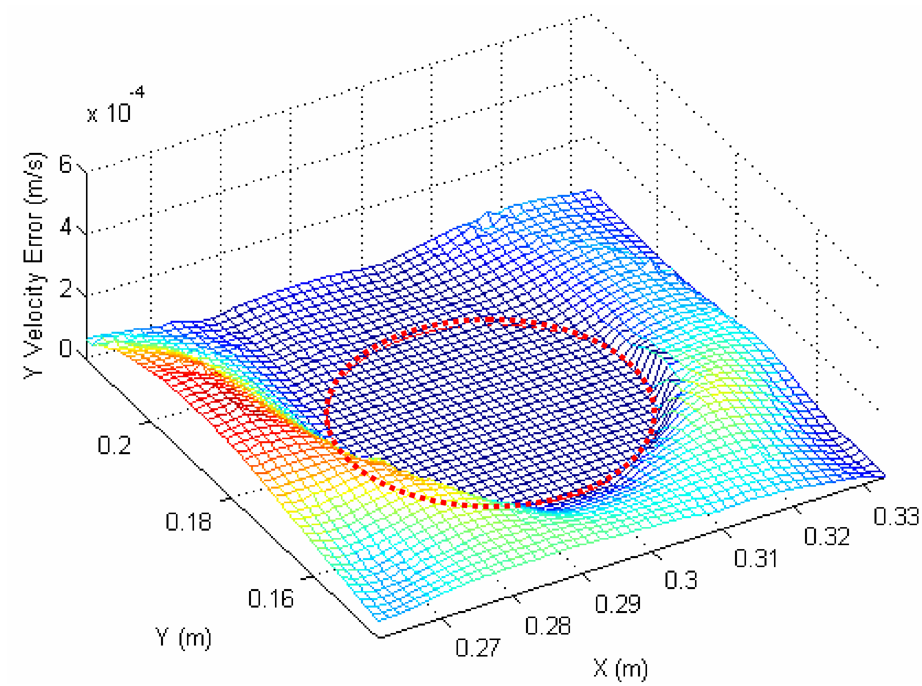


Figure C.42 – Absolute Y-Velocity Error for the ANN2-RF3-CIR Case

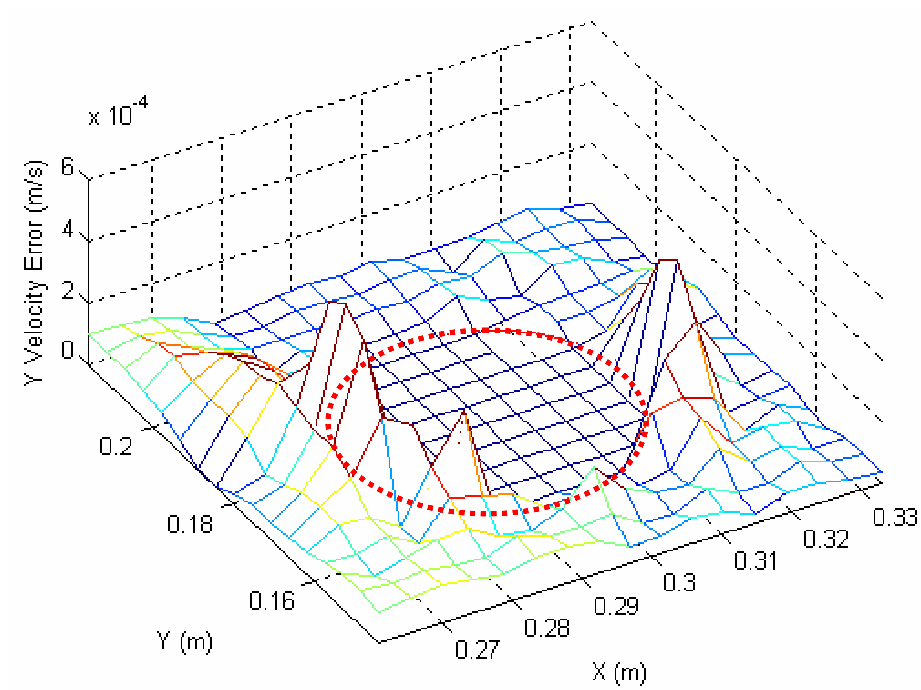


Figure C.43 – Absolute Y-Velocity Error for the ANN2-CIR Case

RMS Error Plots for the Refined Annulus Cases

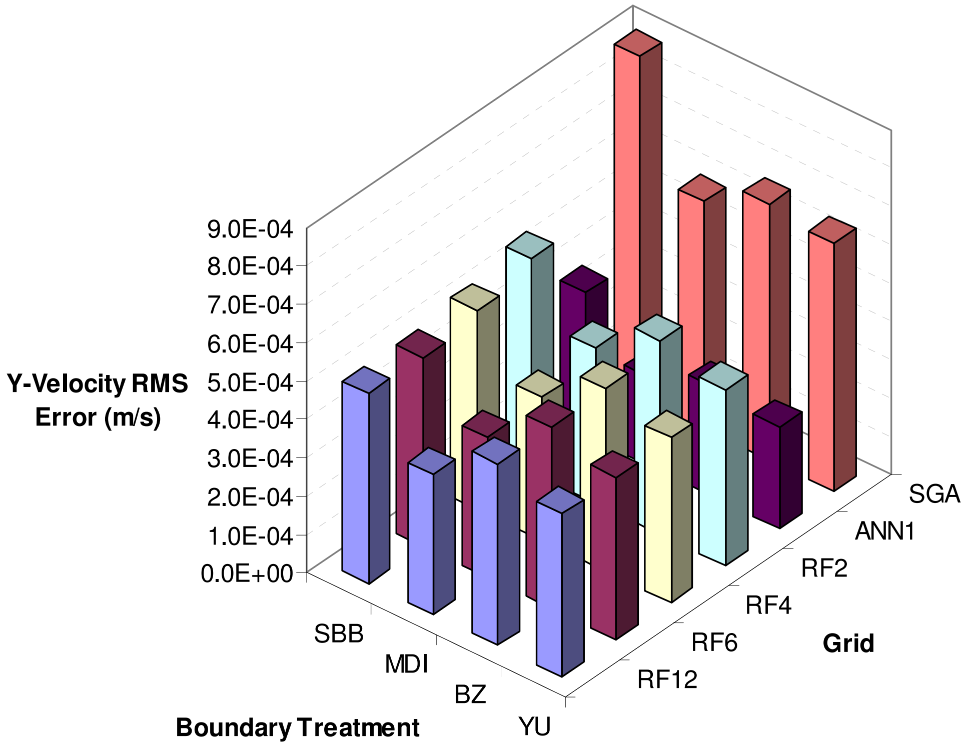


Figure C.44 – Y-Velocity RMS Error for the ANN1-CIR Grid Refinement

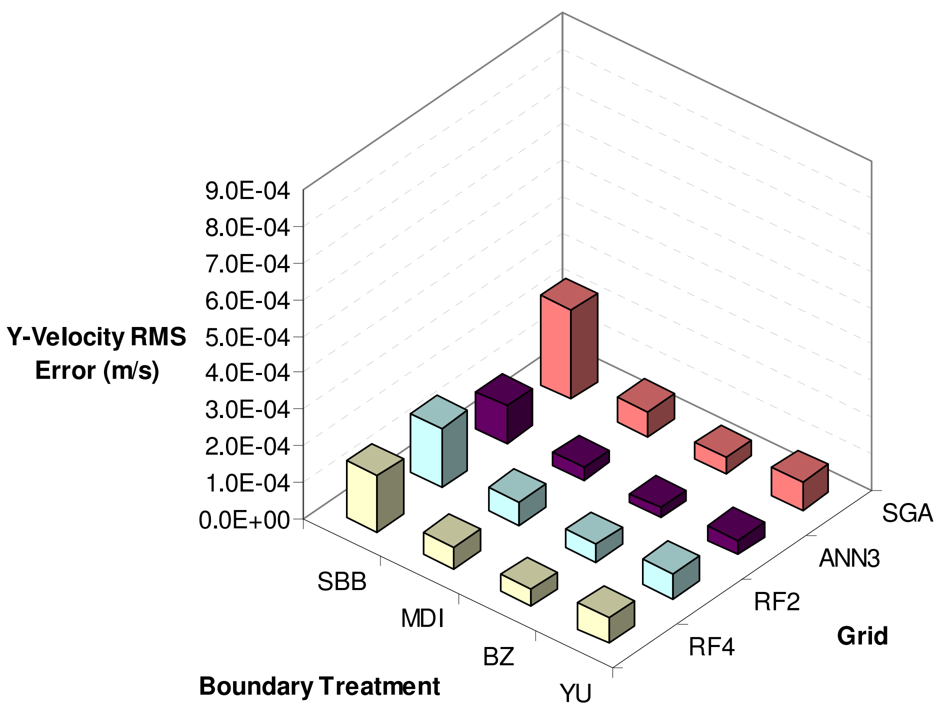


Figure C.45 – Y-Velocity RMS Error for the ANN3-CIR Grid Refinement

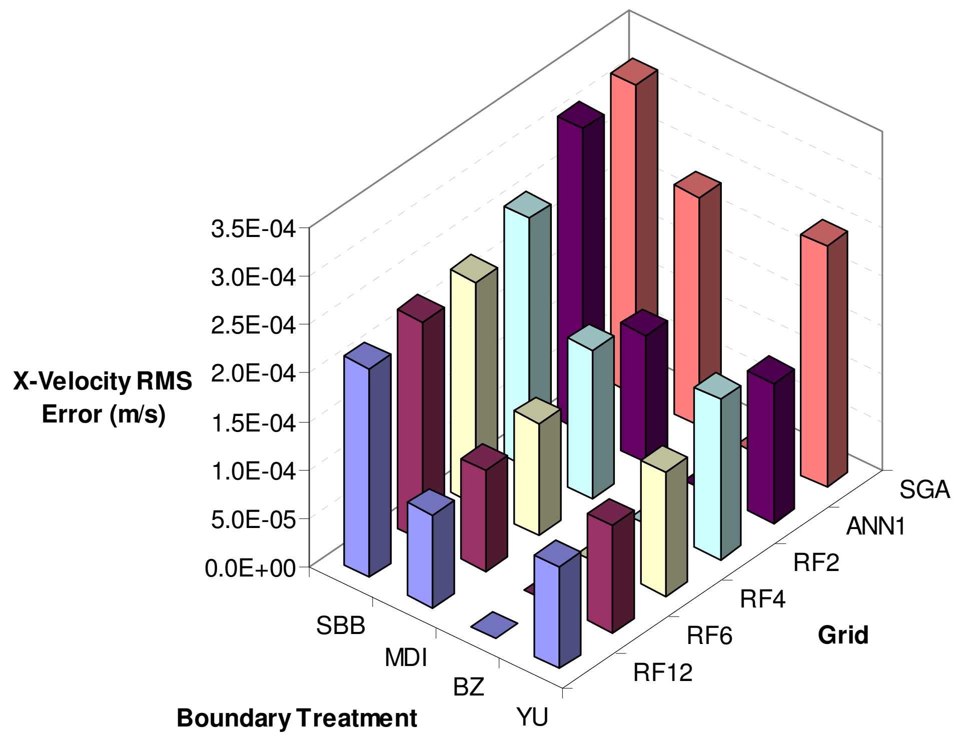


Figure C.46 – X-Velocity RMS Error for the ANN1-SQ Grid Refinement

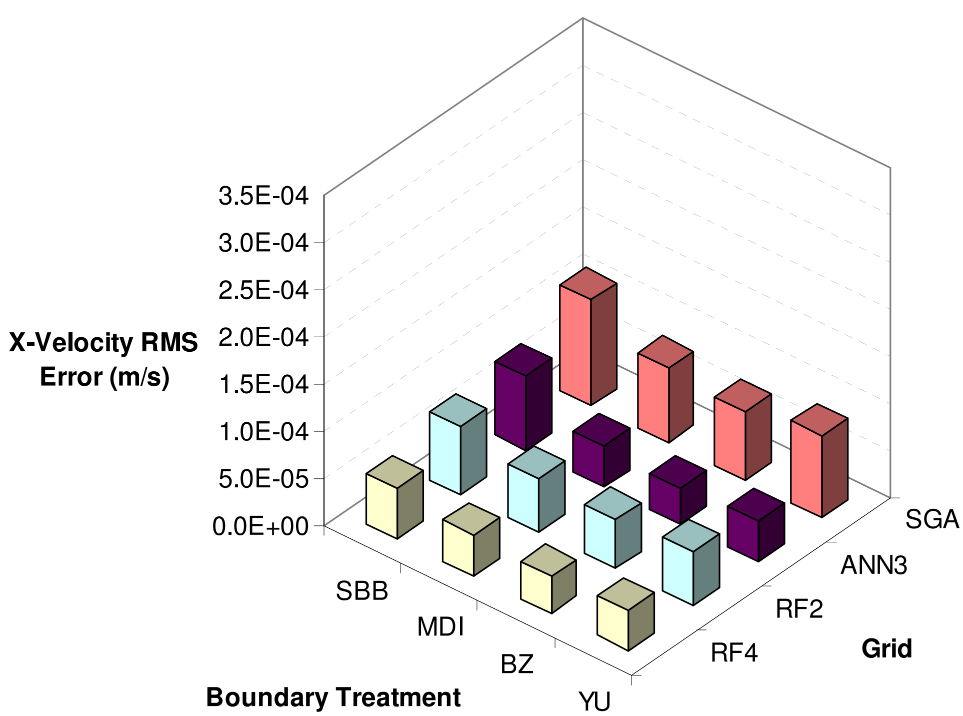


Figure C.47 – X-Velocity RMS Error for the ANN3-SQ Grid Refinement

Absolute Error Results for the Refined Annulus Cases

Table C.18 – Percent Decrease in Y-Velocity RMS Error for ANN-CIR Cases

Grid	Boundary Treatment	RMS error Y-Velocity Sub-Grid (m/s)	RMS error Y-Velocity SGA (m/s)	% Decrease in RMS error
ANN1-CIR RF12	SBB	0.000494	0.000890	44.47%
	MDI	0.000365	0.000594	38.50%
	BZ	0.000474	0.000669	29.09%
	YU	0.000424	0.000648	34.47%
ANN2-CIR RF6	SBB	0.000154	0.000295	47.86%
	MDI	0.000077	0.000133	42.32%
	BZ	0.000082	0.000127	35.24%
	YU	0.000090	0.000142	36.95%
ANN3-CIR RF4	SBB	0.000153	0.000242	36.78%
	MDI	0.000061	0.000069	11.61%
	BZ	0.000045	0.000044	-3.15%
	YU	0.000069	0.000078	11.68%

Table C.19 – Percent Decrease in X-Velocity RMS Error for ANN-SQ Cases

Grid	Boundary Treatment	RMS error X-Velocity Sub-Grid (m/s)	RMS error X-Velocity SGA (m/s)	% Decrease in RMS error
ANN1-SQ RF12	SBB	0.000215	0.000320	32.74%
	MDI	0.000096	0.000236	59.23%
	BZ	N/A	N/A	N/A
	YU	0.000105	0.000249	57.78%
ANN2-SQ RF6	SBB	0.000094	0.000166	43.37%
	MDI	0.000039	0.000132	70.30%
	BZ	0.000035	0.000128	72.87%
	YU	0.000042	0.000135	68.64%
ANN3-SQ RF4	SBB	0.000053	0.000112	52.61%
	MDI	0.000044	0.000079	44.76%
	BZ	0.000038	0.000072	46.67%
	YU	0.000042	0.000087	51.93%

Percent Error Results for the Refined Annulus Cases

Table C.20 – Y-Velocity Percent Error Results for the ANN-CIR Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-CIR RF12	SBB	0.99	5.03	27.67
	MDI	0.33	10.58	63.07
	BZ	0.31	5.40	41.25
	YU	0.27	5.43	47.91
ANN2-CIR RF6	SBB	23.15	57.11	83.48
	MDI	22.36	92.44	98.05
	BZ	12.34	91.77	97.99
	YU	21.62	89.00	98.40
ANN3-CIR RF4	SBB	7.52	56.52	80.88
	MDI	22.64	88.42	95.16
	BZ	41.57	90.89	96.21
	YU	19.88	85.14	94.36

Table C.21 – Y-Velocity Percent Error Results for the ANN-CIR SGA Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-CIR SGA	SBB	3.13	17.19	31.25
	MDI	1.56	20.31	37.50
	BZ	0.00	15.63	28.13
	YU	1.56	15.63	39.06
ANN2-CIR SGA	SBB	16.07	54.46	66.52
	MDI	25.89	70.98	82.14
	BZ	22.32	74.55	83.04
	YU	32.14	70.09	84.38
ANN3-CIR SGA	SBB	13.31	50.52	71.93
	MDI	32.85	87.32	93.56
	BZ	51.56	88.98	93.97
	YU	31.81	84.82	92.52

Table C.22 – X-Velocity Percent Error Results for the ANN-SQ Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-SQ RF12	SBB	1.46	8.94	32.03
	MDI	11.22	34.55	57.32
	BZ	N/A	N/A	N/A
	YU	5.80	27.36	54.40
ANN2-SQ RF6	SBB	11.39	38.95	61.81
	MDI	12.60	52.65	75.70
	BZ	14.16	57.23	76.77
	YU	14.41	48.70	70.66
ANN3-SQ RF4	SBB	11.37	47.76	61.70
	MDI	11.22	57.64	74.96
	BZ	13.01	61.08	75.22
	YU	11.34	59.38	72.09

Table C.23 – X-Velocity Percent Error Results for the ANN-SQ SGA Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-SQ SGA	SBB	1.79	10.71	19.64
	MDI	3.57	14.29	33.93
	BZ	N/A	N/A	N/A
	YU	1.79	10.71	30.36
ANN2-SQ SGA	SBB	3.40	21.85	41.26
	MDI	5.34	31.07	54.37
	BZ	7.77	31.07	53.88
	YU	5.83	31.07	48.06
ANN3-SQ SGA	SBB	9.52	38.78	61.45
	MDI	16.33	55.78	73.47
	BZ	19.05	55.10	72.56
	YU	17.01	51.02	71.20

Table C.24 – Percent Decrease in Y-Velocity RMS Error for ANN-SQ-RF4 Flow Parameter Cases

Grid	Boundary Treatment	RMS error Y-Velocity Sub-Grid (m/s)	RMS error Y-Velocity SGA (m/s)	% Decrease in RMS error
ANN3-SQ-RF4 CAS5	SBB	0.000053	0.000112	52.61%
	MDI	0.000044	0.000079	44.76%
	BZ	0.000038	0.000072	46.67%
	YU	0.000042	0.000087	51.93%
ANN3-SQ-RF4 CAS6	SBB	0.000056	0.000101	44.47%
	MDI	0.000048	0.000082	41.01%
	BZ	0.000042	0.000072	41.09%
	YU	0.000046	0.000074	38.63%
ANN3-SQ-RF4 CAS8	SBB	0.000147	0.000240	38.94%
	MDI	0.000134	0.000197	32.34%
	BZ	0.000111	0.000158	29.68%
	YU	0.000121	0.000182	33.39%

Table C.25 – X-Velocity Percent Error Results for the ANN-SQ-RF4 Flow Parameter Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN3-SQ-RF4 CAS5	SBB	11.37	47.76	61.70
	MDI	11.22	57.64	74.96
	BZ	13.01	61.08	75.22
	YU	11.34	59.38	72.09
ANN3-SQ-RF4 CAS6	SBB	6.39	39.40	61.05
	MDI	10.12	55.00	75.70
	BZ	10.95	60.43	79.34
	YU	10.75	57.44	75.81
ANN3-SQ-RF4 CAS8	SBB	8.48	36.40	58.07
	MDI	14.21	51.38	69.03
	BZ	14.78	57.33	74.02
	YU	14.10	50.53	70.78

Table C.26 – X-Velocity Percent Error Results for the ANN-SQ-RF4 SGA Flow Parameter Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN3-SQ-RF4 CAS5 SGA	SBB	9.52	38.78	61.45
	MDI	16.33	55.78	73.47
	BZ	19.05	55.10	72.56
	YU	17.01	51.02	71.20
ANN3-SQ-RF4 CAS6 SGA	SBB	9.26	40.18	59.59
	MDI	12.19	47.40	68.17
	BZ	11.96	50.56	69.07
	YU	16.03	52.37	72.01
ANN3-SQ-RF4 CAS8 SGA	SBB	7.00	39.05	59.82
	MDI	12.87	47.86	67.27
	BZ	16.93	52.14	71.33
	YU	14.22	52.82	68.62

RMS Error Results for the Mixed Boundary Treatment Cases

Table C.27 – Percent Decrease in X-Velocity Error for the ANN3-EL Mix Cases

Grid	Main Grid Boundary Treatment	Sub-Grid Boundary Treatment	RMS error X-Velocity Sub-Grid (m/s)	RMS error X-Velocity SGA (m/s)	% Decrease in RMS error
ANN3-EL RF4	SBB	SBB	0.000063	0.000125	49.92%
		MDI	0.000061	0.000125	51.27%
		BZ	0.000061	0.000125	51.50%
		YU	0.000061	0.000125	51.58%
ANN3-EL RF4	MDI	SBB	0.000030	0.000051	40.63%
		MDI	0.000024	0.000051	51.86%
		BZ	0.000024	0.000051	52.04%
		YU	0.000024	0.000051	52.74%
ANN3-EL RF4	BZ	SBB	0.000030	0.000050	40.74%
		MDI	0.000025	0.000050	51.10%
		BZ	0.000024	0.000050	51.57%
		YU	0.000024	0.000050	51.93%
ANN3-EL RF4	YU	SBB	0.000034	0.000060	44.53%
		MDI	0.000029	0.000060	51.31%
		BZ	0.000029	0.000060	51.66%
		YU	0.000029	0.000060	52.12%

Table C.28 – Percent Decrease in Y-Velocity Error for the ANN3-EL Mix Cases

Grid	Main Grid Boundary Treatment	Sub-Grid Boundary Treatment	RMS error Y-Velocity Sub-Grid (m/s)	RMS error Y-Velocity SGA (m/s)	% Decrease in RMS error
ANN3-EL RF4	SBB	SBB	0.000120	0.000251	52.24%
		MDI	0.000110	0.000251	56.10%
		BZ	0.000110	0.000251	56.33%
		YU	0.000111	0.000251	55.68%
ANN3-EL RF4	MDI	SBB	0.000057	0.000084	31.24%
		MDI	0.000045	0.000084	45.58%
		BZ	0.000045	0.000084	46.23%
		YU	0.000046	0.000084	44.67%
ANN3-EL RF4	BZ	SBB	0.000074	0.000097	24.22%
		MDI	0.000064	0.000097	34.32%
		BZ	0.000063	0.000097	34.91%
		YU	0.000064	0.000097	33.77%
ANN3-EL RF4	YU	SBB	0.000076	0.000115	33.89%
		MDI	0.000066	0.000115	42.68%
		BZ	0.000066	0.000115	43.22%
		YU	0.000067	0.000115	42.12%

Force Evaluation on a Stationary Construct

<u>Force and Moment Results for Main Grid Cases</u>

Table C.29 – Force Results for the ANN-SQ Main Grid Cases

Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000061		0.002549		0.000034	
ANN1-SQ	SBB	-0.000074	21.40%	0.002976	16.73%	0.000008	76.40%
	MDI	-0.000123	102.70%	0.002456	3.68%	0.000126	270.86%
	BZ	N/A	N/A	N/A	N/A	N/A	N/A
	YU	-0.000108	77.85%	0.002538	0.43%	0.000120	255.24%
ANN2-SQ	SBB	-0.000017	72.64%	0.002572	0.89%	0.000005	86.09%
	MDI	-0.000103	69.22%	0.002514	1.37%	0.000042	22.70%
	BZ	-0.000001	97.97%	0.002603	2.12%	0.000016	54.17%
	YU	-0.000035	43.01%	0.002630	3.16%	0.000060	77.87%
ANN3-SQ	SBB	-0.000064	5.88%	0.002637	3.45%	0.000046	34.69%
	MDI	-0.000032	47.55%	0.002510	1.55%	0.000045	33.34%
	BZ	-0.000086	41.02%	0.002549	0.01%	0.000035	3.18%
	YU	-0.000040	34.44%	0.002438	4.35%	0.000021	37.33%

Table C.30 – Force Results for the ANN-CIR Main Grid Cases

Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		0.000145		0.002218		0.000019	
ANN1-CIR	SBB	0.000312	115.59%	0.002640	19.00%	0.000062	231.68%
	MDI	0.000100	31.16%	0.002862	29.02%	-0.000093	596.93%
	BZ	0.000227	56.44%	0.002573	15.98%	0.000063	235.52%
	YU	-0.000288	299.06%	0.002349	5.87%	-0.000206	1200.97%
ANN2-CIR	SBB	0.000177	22.47%	0.002341	5.53%	0.000010	44.15%
	MDI	0.000111	23.49%	0.002412	8.74%	0.000017	8.04%
	BZ	0.000134	7.56%	0.002281	2.84%	-0.000006	131.52%
	YU	0.000089	38.40%	0.002290	3.25%	0.000053	182.28%
ANN3-CIR	SBB	0.000098	32.36%	0.002335	5.24%	0.000012	34.50%
	MDI	0.000148	2.36%	0.002270	2.31%	0.000038	102.32%
	BZ	0.000085	40.97%	0.002328	4.93%	0.000031	63.12%
	YU	0.000172	18.42%	0.002176	1.92%	0.000032	69.79%

Table C.31 – Force Results for the ANN-RECT Main Grid Cases

Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000063		0.002384		-0.000012	
ANN1-RECT	SBB	-0.000236	273.15%	0.002667	11.87%	0.000002	117.44%
	MDI	-0.000042	33.71%	0.002851	19.60%	-0.000113	829.12%
	BZ	-0.000237	274.32%	0.002479	4.00%	-0.000011	13.63%
	YU	-0.000123	94.43%	0.002956	24.01%	-0.000065	437.56%
ANN2-RECT	SBB	-0.000049	21.92%	0.002549	6.92%	0.000020	264.27%
	MDI	-0.000004	93.01%	0.002339	1.88%	0.000025	307.62%
	BZ	0.000024	137.46%	0.002354	1.25%	0.000058	579.00%
	YU	-0.000187	196.19%	0.002490	4.46%	-0.000004	69.32%
ANN3-RECT	SBB	-0.000206	224.88%	0.002383	0.03%	-0.000004	64.96%
	MDI	-0.000194	207.35%	0.002388	0.17%	-0.000025	109.19%
	BZ	-0.000207	226.82%	0.002265	5.00%	-0.000021	71.35%
	YU	-0.000333	426.73%	0.002373	0.48%	-0.000024	95.37%

Table C.32 – Force Results for the ANN-EL Main Grid Cases

Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000262		0.002140		-0.000019	
ANN1-EL	SBB	-0.000169	35.35%	0.002265	5.83%	0.000029	258.04%
	MDI	-0.000834	218.55%	0.003055	42.72%	0.000070	474.46%
	BZ	0.000154	158.75%	0.002118	1.05%	0.000035	288.44%
	YU	-0.000311	18.85%	0.002244	4.86%	-0.000050	166.04%
ANN2-EL	SBB	-0.000204	22.10%	0.002251	5.19%	-0.000030	59.12%
	MDI	-0.000248	5.33%	0.002289	6.94%	-0.000023	25.12%
	BZ	-0.000170	35.09%	0.002078	2.93%	-0.000020	7.03%
	YU	-0.000161	38.49%	0.002212	3.33%	-0.000008	57.74%
ANN3-EL	SBB	-0.000218	16.71%	0.002167	1.26%	-0.000038	101.76%
	MDI	-0.000243	7.07%	0.002273	6.21%	-0.000033	78.13%
	BZ	-0.000312	19.32%	0.002223	3.86%	-0.000042	124.69%
	YU	-0.000202	22.98%	0.002167	1.25%	-0.000006	65.31%

Force and Moment Results for Sub-Grid Cases

Table C.33 – Force Results for the ANN-SQ Sub-Grid Cases

Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000061		0.002549		0.000034	
ANN1-SQ RF12	SBB	-0.000010	83.27%	0.001905	25.27%	0.000036	5.28%
	MDI	-0.000050	16.92%	0.002322	8.90%	0.000039	16.00%
	BZ	N/A	N/A	N/A	N/A	N/A	N/A
	YU	-0.000035	41.76%	0.002293	10.04%	0.000038	12.46%
ANN2-SQ RF6	SBB	0.000009	114.49%	0.002617	2.65%	0.000043	27.93%
	MDI	-0.000023	61.64%	0.002539	0.40%	0.000042	23.91%
	BZ	-0.000020	67.54%	0.002518	1.22%	0.000038	12.66%
	YU	-0.000016	73.92%	0.002597	1.88%	0.000042	23.83%
ANN3-SQ RF4	SBB	-0.000088	45.17%	0.002373	6.91%	0.000032	4.58%
	MDI	-0.000072	18.45%	0.002417	5.17%	0.000035	3.66%
	BZ	-0.000079	30.32%	0.002404	5.68%	0.000033	3.93%
	YU	-0.000075	23.42%	0.002440	4.29%	0.000035	2.71%

Table C.34 – Force Results for the ANN-CIR Sub-Grid Cases

Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		0.000145		0.002218		0.000019	
ANN1-CIR RF12	SBB	0.000120	17.00%	0.001803	18.71%	0.000018	6.49%
	MDI	0.000124	14.24%	0.001940	12.54%	0.000019	0.89%
	BZ	0.000129	10.62%	0.001858	16.25%	0.000018	4.41%
	YU	0.000134	7.73%	0.001889	14.84%	0.000019	0.25%
ANN2-CIR RF6	SBB	0.000159	9.66%	0.002066	6.85%	0.000019	3.38%
	MDI	0.000141	2.33%	0.002128	4.08%	0.000019	1.46%
	BZ	0.000143	0.99%	0.002116	4.63%	0.000018	3.13%
	YU	0.000149	3.23%	0.002120	4.41%	0.000019	3.22%
ANN3-CIR RF4	SBB	0.000137	5.19%	0.002047	7.72%	0.000017	10.97%
	MDI	0.000133	8.46%	0.002147	3.22%	0.000018	2.44%
	BZ	0.000136	5.87%	0.002159	2.68%	0.000018	4.41%
	YU	0.000135	6.89%	0.002141	3.49%	0.000018	2.87%

Table C.35 – Force Results for the ANN-RECT Sub-Grid Cases

Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000063		0.002384		-0.000012	
ANN1-RECT RF12	SBB	-0.000064	1.27%	0.002294	3.76%	-0.000008	32.79%
	MDI	-0.000064	0.91%	0.002181	8.52%	-0.000007	45.96%
	BZ	-0.000057	10.32%	0.002151	9.78%	-0.000005	61.83%
	YU	-0.000050	21.16%	0.002215	7.09%	-0.000008	31.72%
ANN2-RECT RF6	SBB	-0.000070	9.85%	0.002203	7.59%	-0.000011	13.13%
	MDI	-0.000044	30.22%	0.002321	2.63%	-0.000012	2.08%
	BZ	-0.000044	30.90%	0.002364	0.84%	-0.000011	9.84%
	YU	-0.000049	23.14%	0.002330	2.28%	-0.000013	5.70%
ANN3-RECT RF4	SBB	-0.000113	78.09%	0.002567	7.66%	-0.000014	12.70%
	MDI	-0.000078	23.43%	0.002508	5.20%	-0.000014	14.09%
	BZ	-0.000075	18.91%	0.002503	5.01%	-0.000012	2.52%
	YU	-0.000075	18.47%	0.002519	5.66%	-0.000013	8.91%

Table C.36 – Force Results for the ANN-EL Sub-Grid Cases

Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000262		0.002140		-0.000019	
ANN1-EL RF12	SBB	-0.000221	15.37%	0.002274	6.24%	-0.000027	42.52%
	MDI	-0.000261	0.22%	0.002266	5.85%	-0.000024	27.00%
	BZ	-0.000246	6.08%	0.002185	2.06%	-0.000025	32.32%
	YU	-0.000234	10.65%	0.002190	2.30%	-0.000021	12.99%
ANN2-EL RF6	SBB	-0.000235	10.26%	0.002026	5.34%	-0.000018	6.02%
	MDI	-0.000244	6.62%	0.002018	5.73%	-0.000017	6.99%
	BZ	-0.000232	11.53%	0.001973	7.83%	-0.000018	2.80%
	YU	-0.000233	11.12%	0.002014	5.90%	-0.000015	21.43%
ANN3-EL RF4	SBB	-0.000223	14.69%	0.002122	0.84%	-0.000017	8.38%
	MDI	-0.000256	2.37%	0.002118	1.06%	-0.000019	0.10%
	BZ	-0.000242	7.40%	0.002097	2.02%	-0.000020	5.20%
	YU	-0.000238	9.02%	0.002098	2.00%	-0.000015	18.48%

Table C.37 – Force Results for the ANN3-EL-RF4 Sub-Grid Flow Parameter Cases

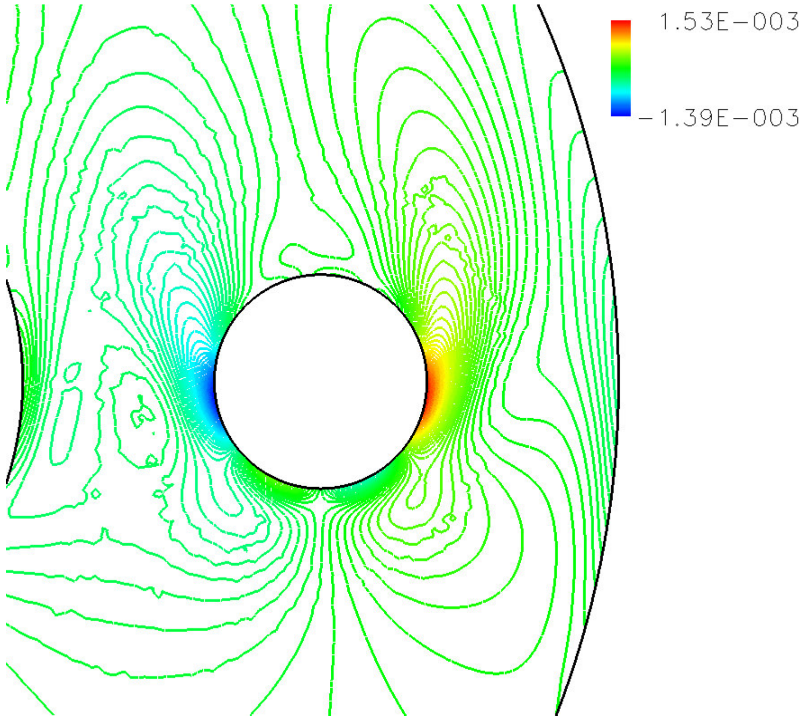
Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000262		0.002140		-0.000019	
ANN3-EL-RF4 CAS5	SBB	-0.000223	14.69%	0.002122	0.84%	-0.000017	8.38%
	MDI	-0.000256	2.37%	0.002118	1.06%	-0.000019	0.10%
	BZ	-0.000242	7.40%	0.002097	2.02%	-0.000020	5.20%
	YU	-0.000238	9.02%	0.002098	2.00%	-0.000015	18.48%
FLUENT		-0.000191		0.001228		-0.000011	
ANN3-EL-RF4 CAS6	SBB	-0.000153	19.81%	0.001229	0.07%	-0.000010	8.94%
	MDI	-0.000187	2.07%	0.001215	1.05%	-0.000010	10.82%
	BZ	-0.000169	11.62%	0.001198	2.47%	-0.000011	1.70%
	YU	-0.000168	12.20%	0.001199	2.39%	-0.000007	37.38%
FLUENT		-0.000766		0.004918		-0.000045	
ANN3-EL-RF4 CAS8	SBB	-0.000718	6.26%	0.004703	4.37%	-0.000040	10.07%
	MDI	-0.000768	0.21%	0.004711	4.21%	-0.000040	10.52%
	BZ	-0.000723	5.57%	0.004629	5.87%	-0.000043	2.98%
	YU	-0.000721	5.95%	0.004648	5.48%	-0.000035	22.41%

Table C.38 – Force Results for the ANN3-EL-RF4 Sub-Grid Mix Cases

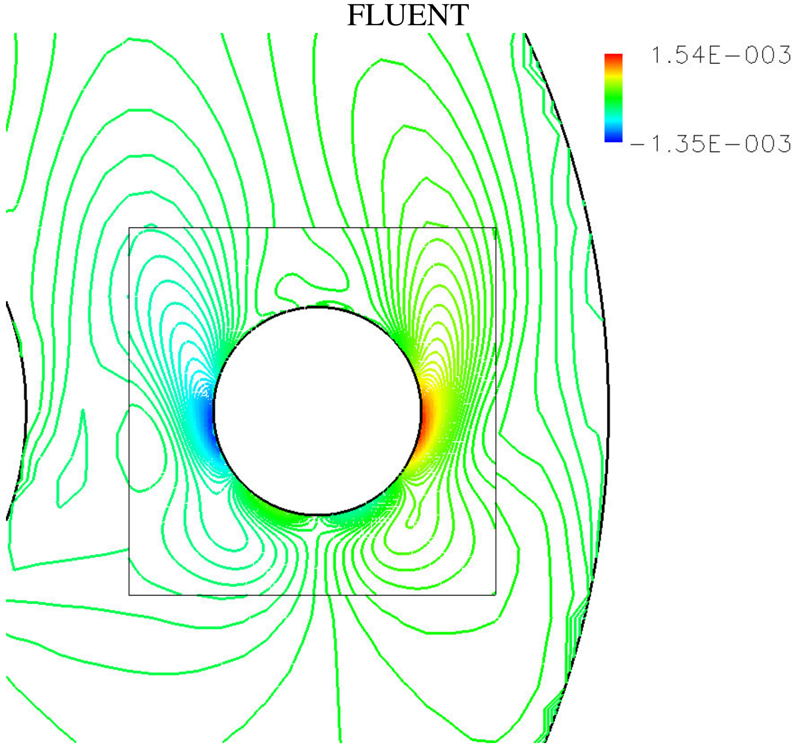
Grid	Sub-Grid Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000262		0.002140		-0.000019	
ANN3-EL-RF4 SBB	SBB	-0.000223	14.69%	0.002122	0.84%	-0.000017	8.38%
	MDI	-0.000222	15.09%	0.002123	0.82%	-0.000016	13.38%
	BZ	-0.000218	16.86%	0.002122	0.87%	-0.000018	5.91%
	YU	-0.000216	17.58%	0.002121	0.92%	-0.000013	27.99%
ANN3-EL-RF4 MDI	SBB	-0.000255	2.37%	0.002118	1.06%	-0.000019	4.61%
	MDI	-0.000256	2.37%	0.002118	1.06%	-0.000019	0.10%
	BZ	-0.000251	4.19%	0.002116	1.12%	-0.000020	7.46%
	YU	-0.000249	4.86%	0.002115	1.18%	-0.000016	14.51%
ANN3-EL-RF4 BZ	SBB	-0.000247	5.62%	0.002098	1.96%	-0.000019	2.36%
	MDI	-0.000247	5.67%	0.002098	1.96%	-0.000018	2.16%
	BZ	-0.000242	7.40%	0.002097	2.02%	-0.000020	5.20%
	YU	-0.000240	8.12%	0.002096	2.07%	-0.000016	16.65%
ANN3-EL-RF4 YU	SBB	-0.000245	6.53%	0.002100	1.89%	-0.000019	0.53%
	MDI	-0.000244	6.62%	0.002100	1.89%	-0.000018	4.04%
	BZ	-0.000240	8.39%	0.002099	1.95%	-0.000019	3.32%
	YU	-0.000238	9.02%	0.002098	2.00%	-0.000015	18.48%

Shear Stress Evaluation for Flow in an Annulus

<u>Contour Plots</u>

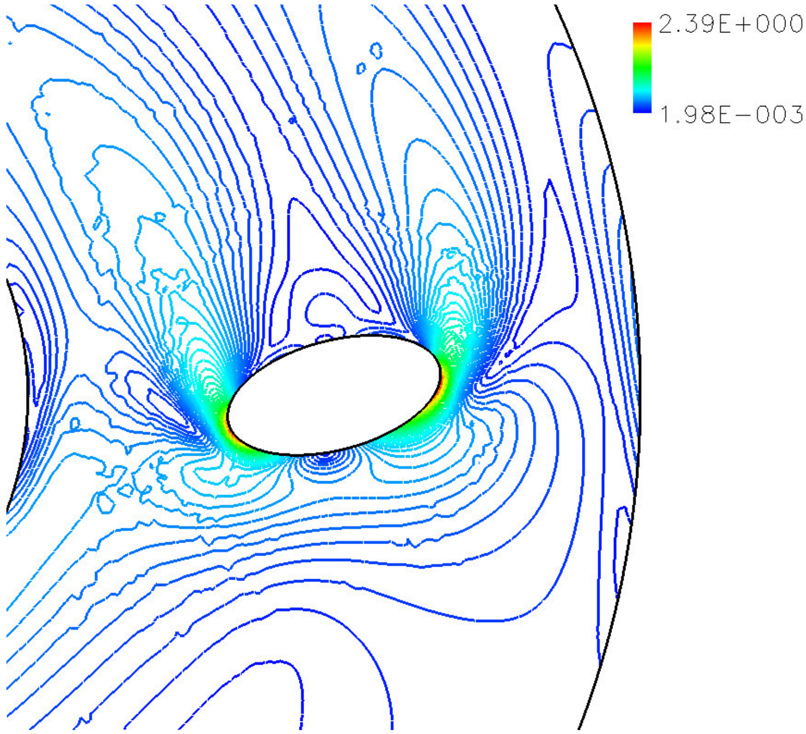


FLUENT

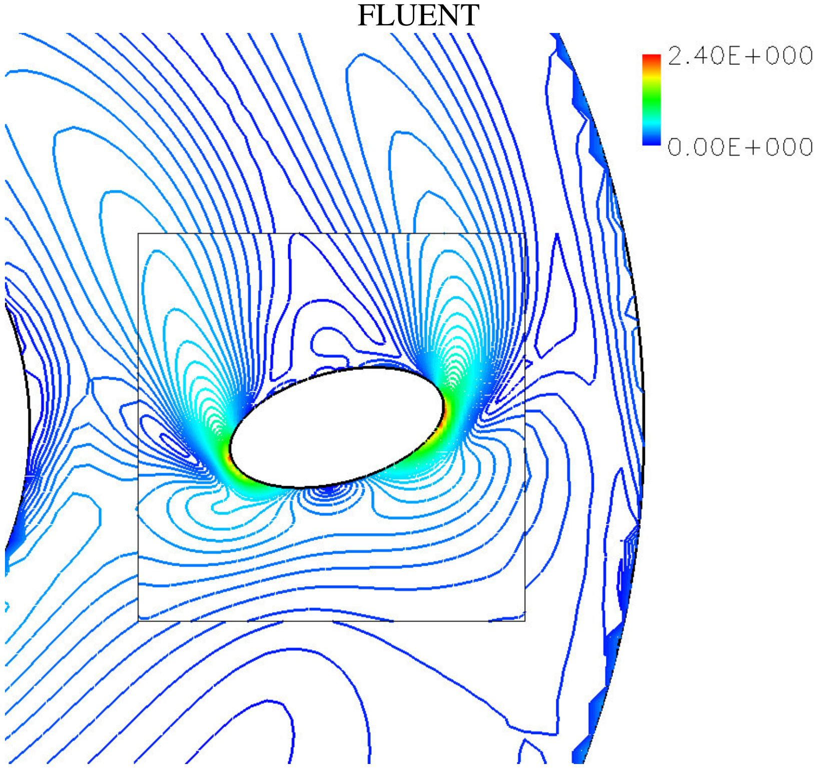


LB

Figure C.48 – Shear Stress Contour Plots for the ANN-CIR Construct Case



FLUENT



LB

Figure C.49 – Shear Rate Contour Plots for the Annulus Case with an EL Construct

Shear Stress and Shear Rate Line Plots

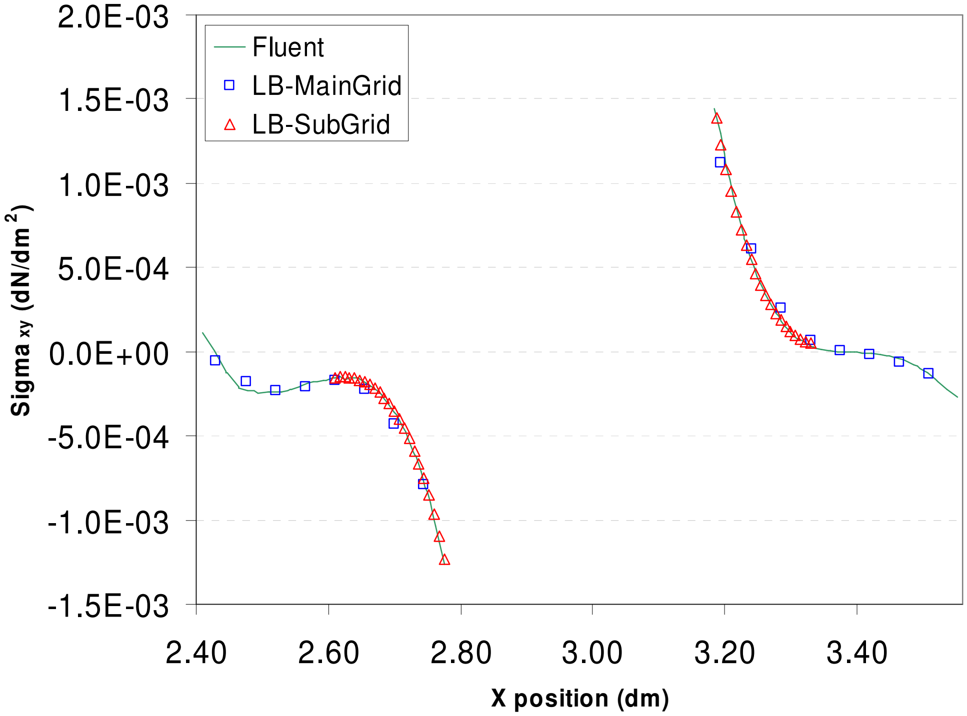


Figure C.50 – Shear Stress Plot Comparing Main Grid and Sub-Grid Values

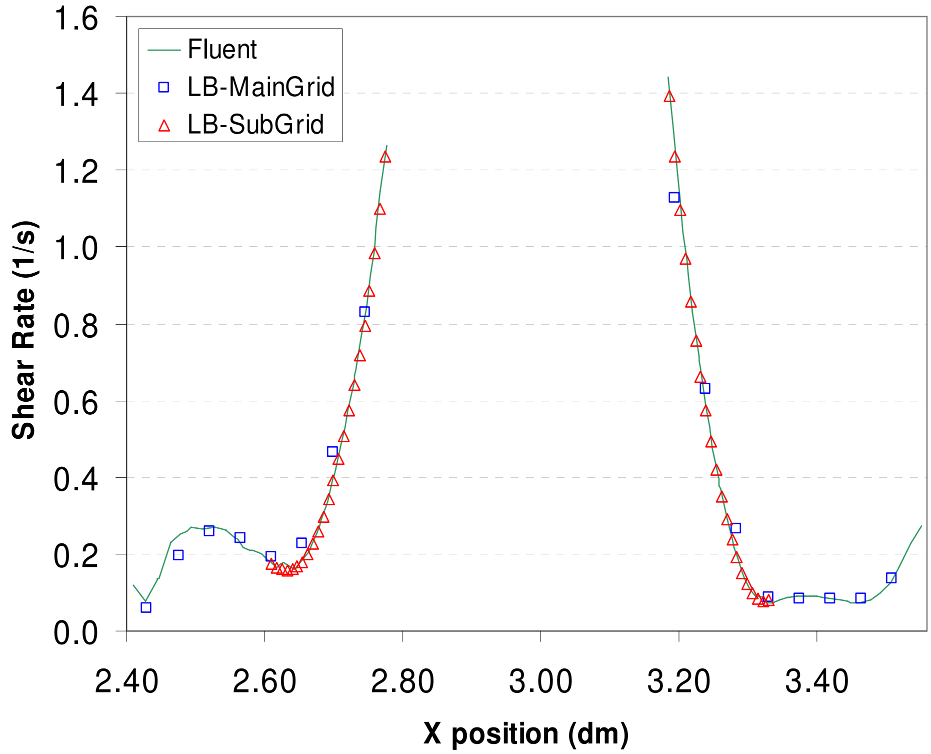


Figure C.51 – Shear Rate Plot Comparing Main Grid and Sub-Grid Values

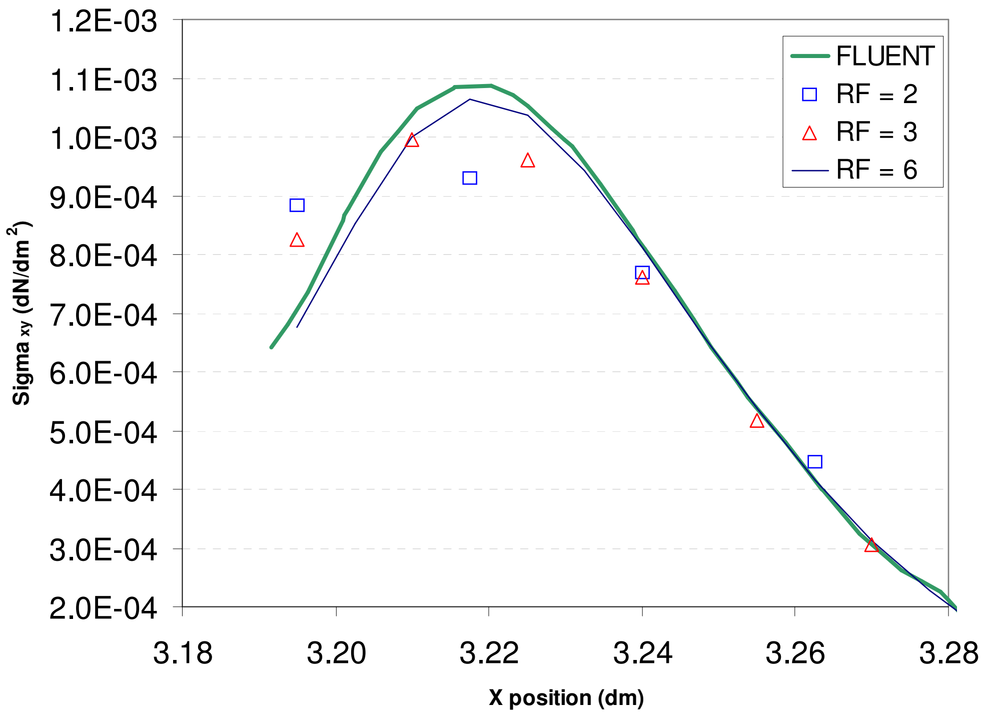


Figure C.52 – Shear Stress Plot Comparing Refinement Factors

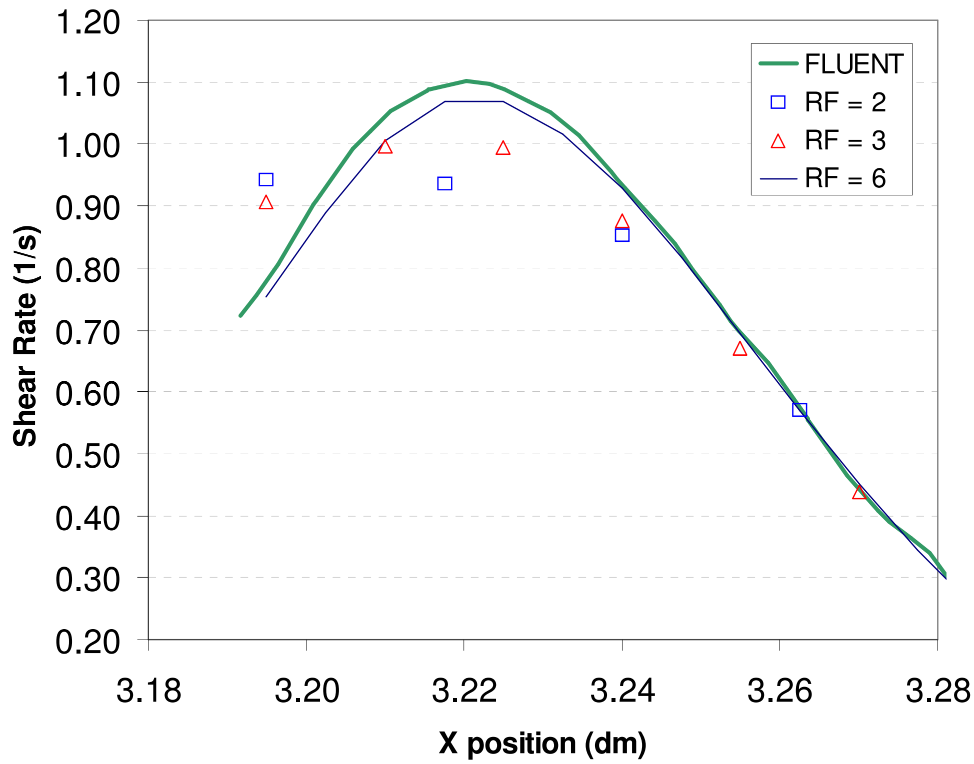


Figure C.53 – Shear Rate Plot Comparing Refinement Factors

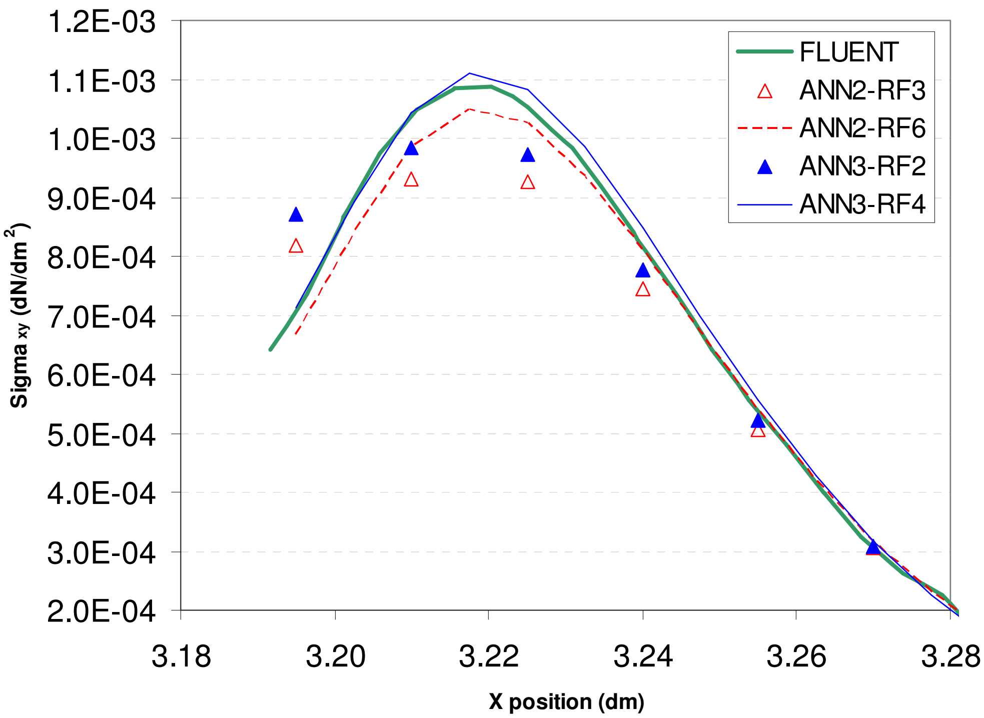


Figure C.54 – Shear Stress Plot Comparing Results across Main Grids

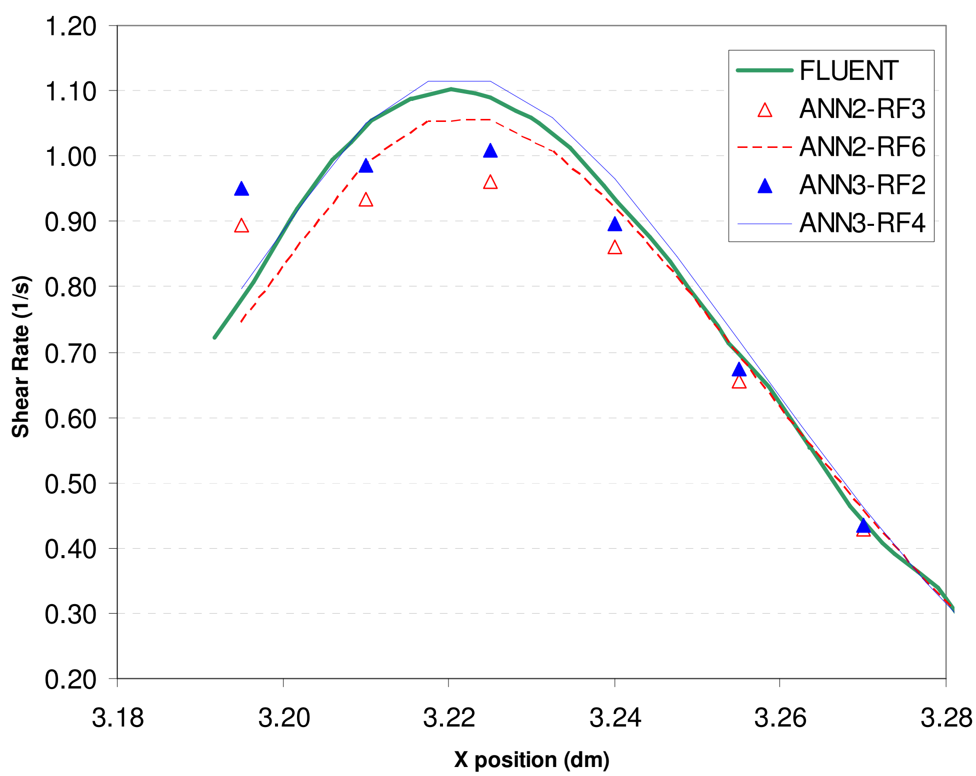


Figure C.55 – Shear Rate Plot Comparing Results across Main Grids

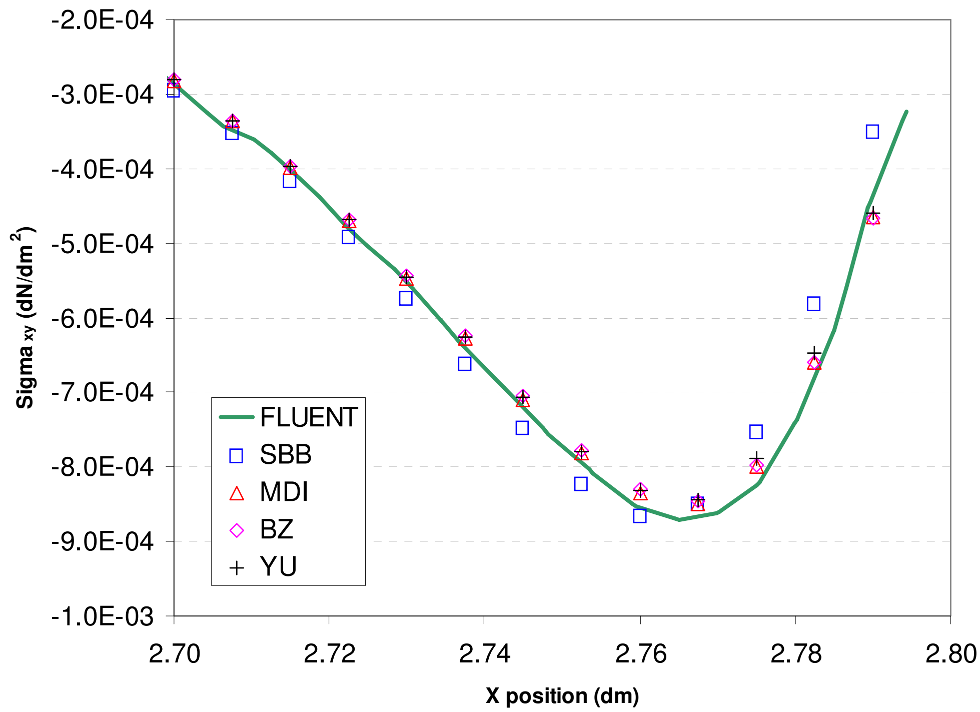


Figure C.56 – Shear Stress Plot Comparing Results across Boundary Treatments

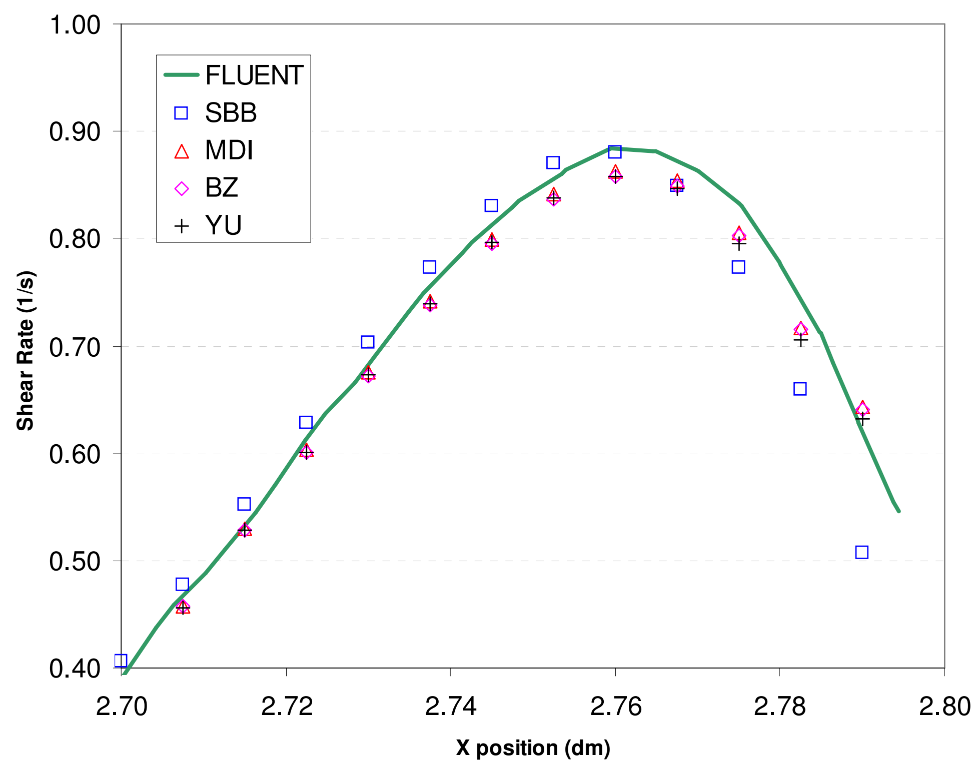


Figure C.57 – Shear Rate Plot Comparing Results across Boundary Treatments

Non-Dimensional Force versus Gap Size Plots

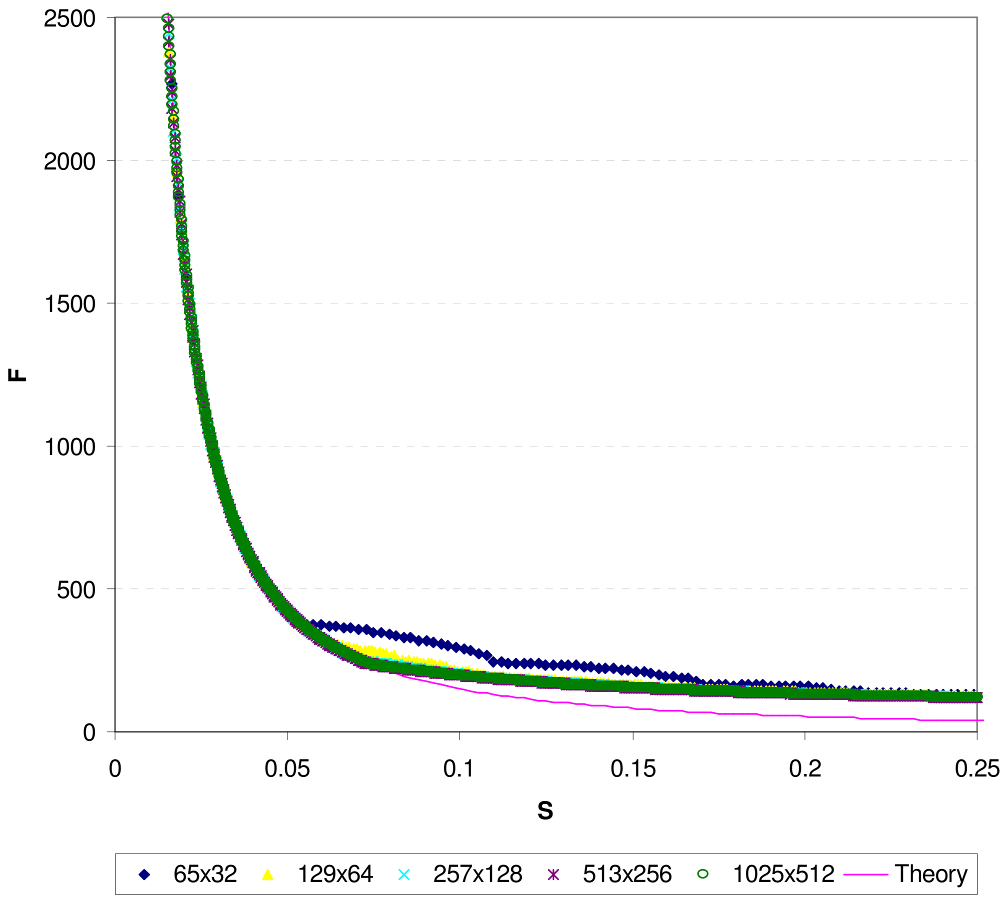


Figure C.58 – Force versus Gap Size for the SBB Boundary Treatment

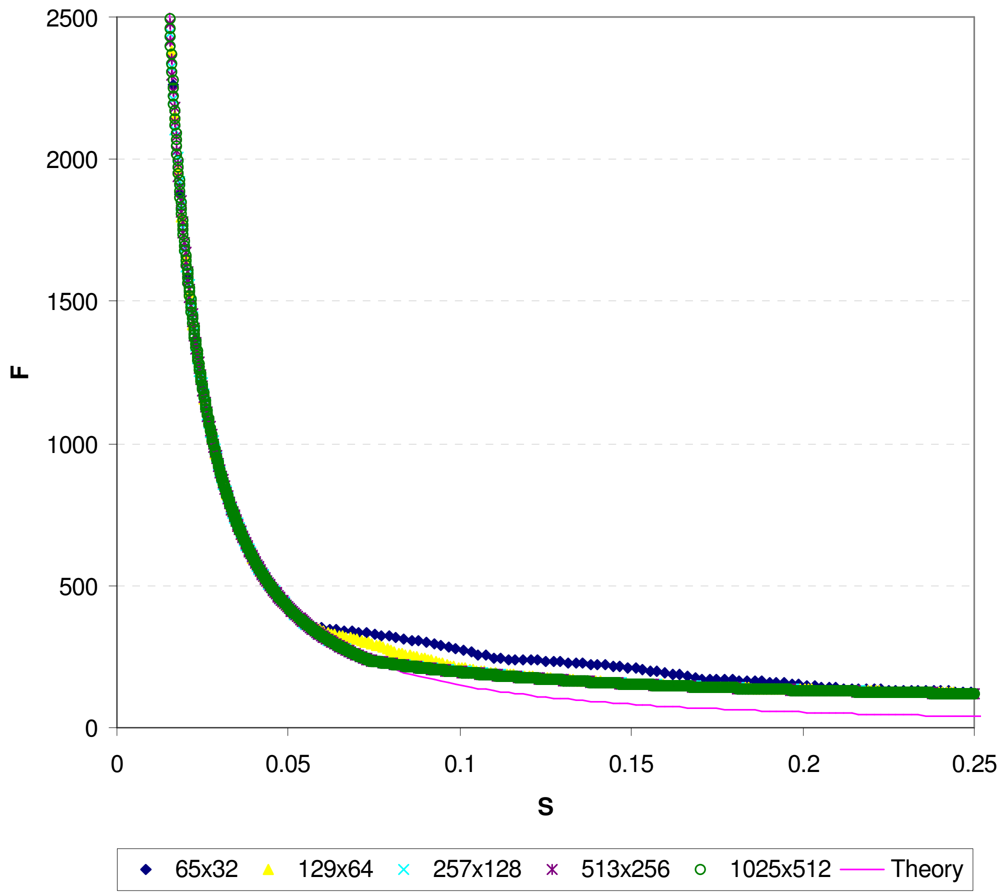


Figure C.59 – Force versus Gap Size for the BZ Boundary Treatment

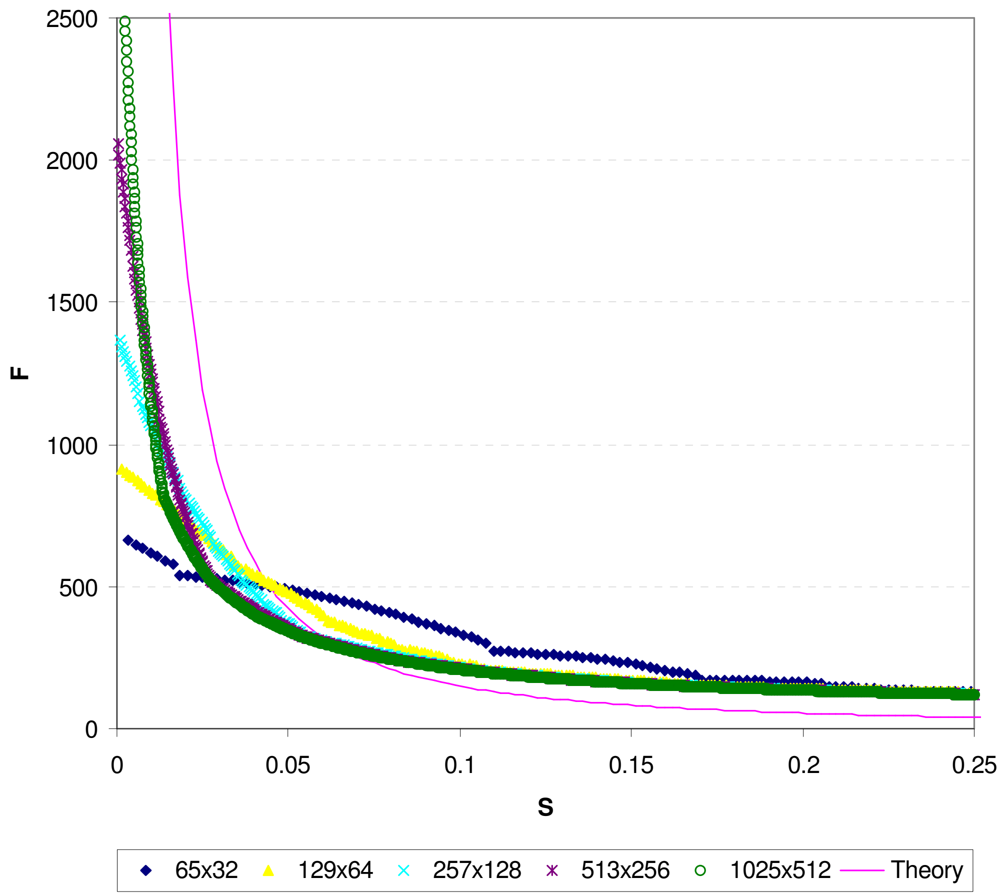


Figure C.60 – Force versus Gap Size for the No Lubrication Force Cases

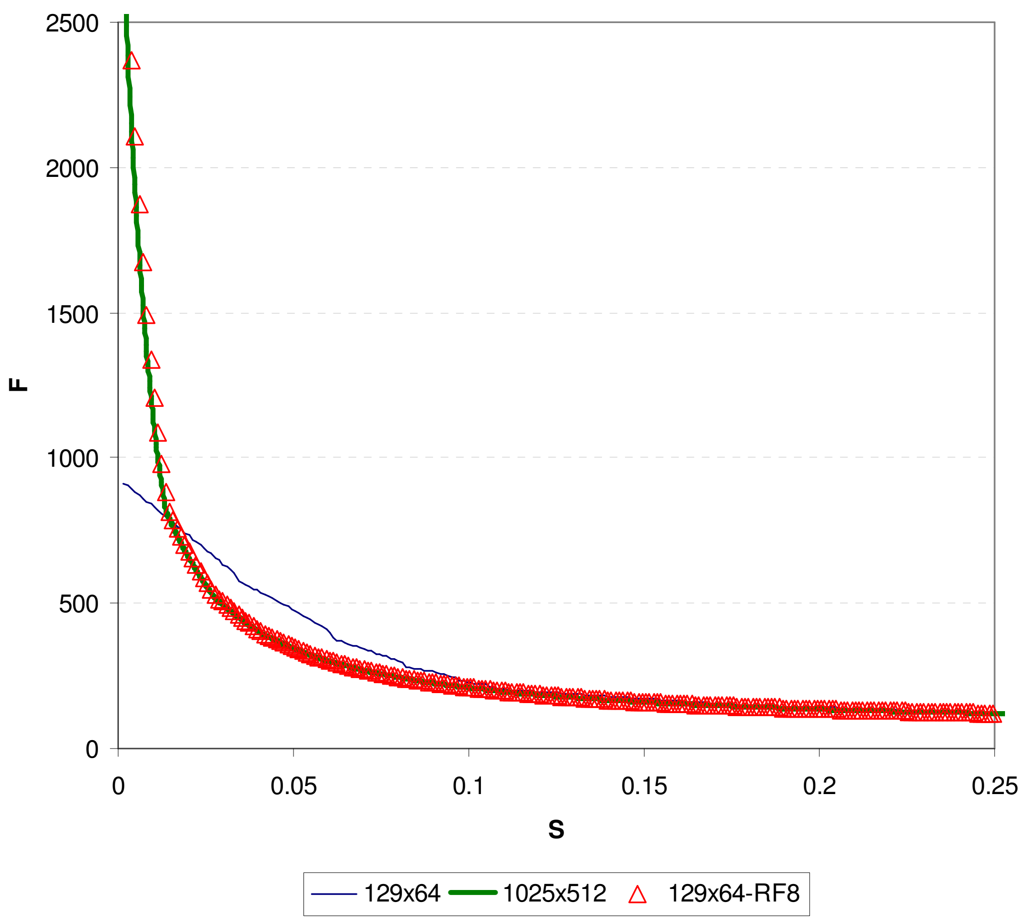


Figure C.61 – Force versus Gap Size for the RF8 Sub-Grid Case

<u>Drag Force versus Channel Geometry Plots</u>

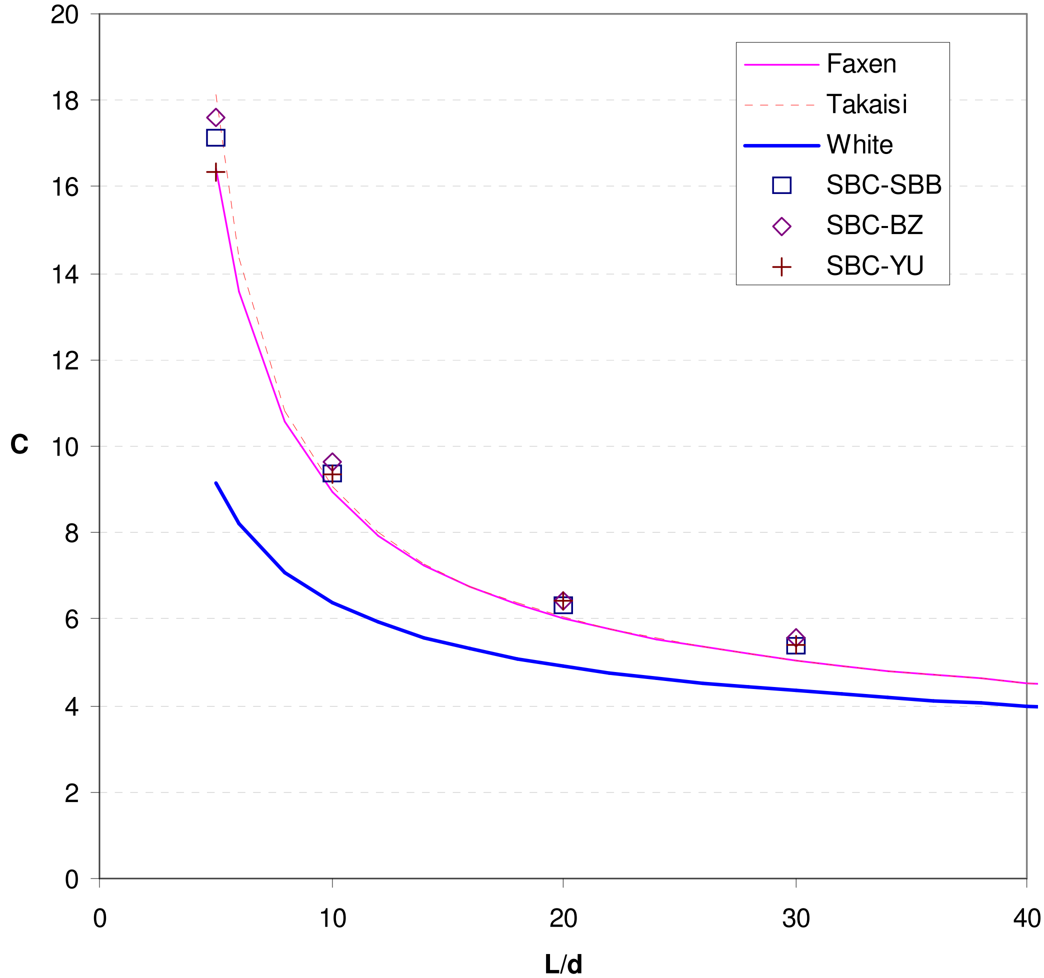


Figure C.62 – C versus L/d for the Stokes Boundary Condition (SBC)

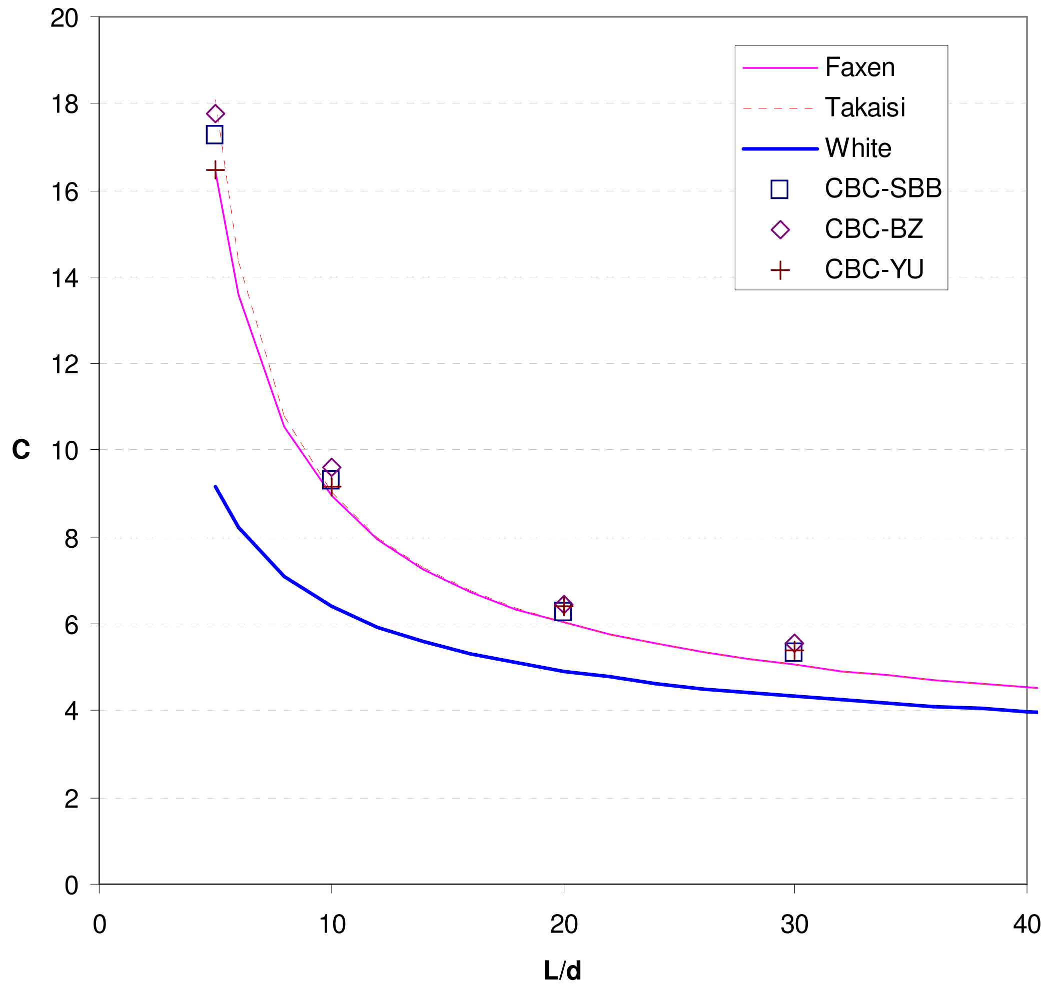


Figure C.63 – C versus L/d for the Cavity Boundary Condition (CBC)

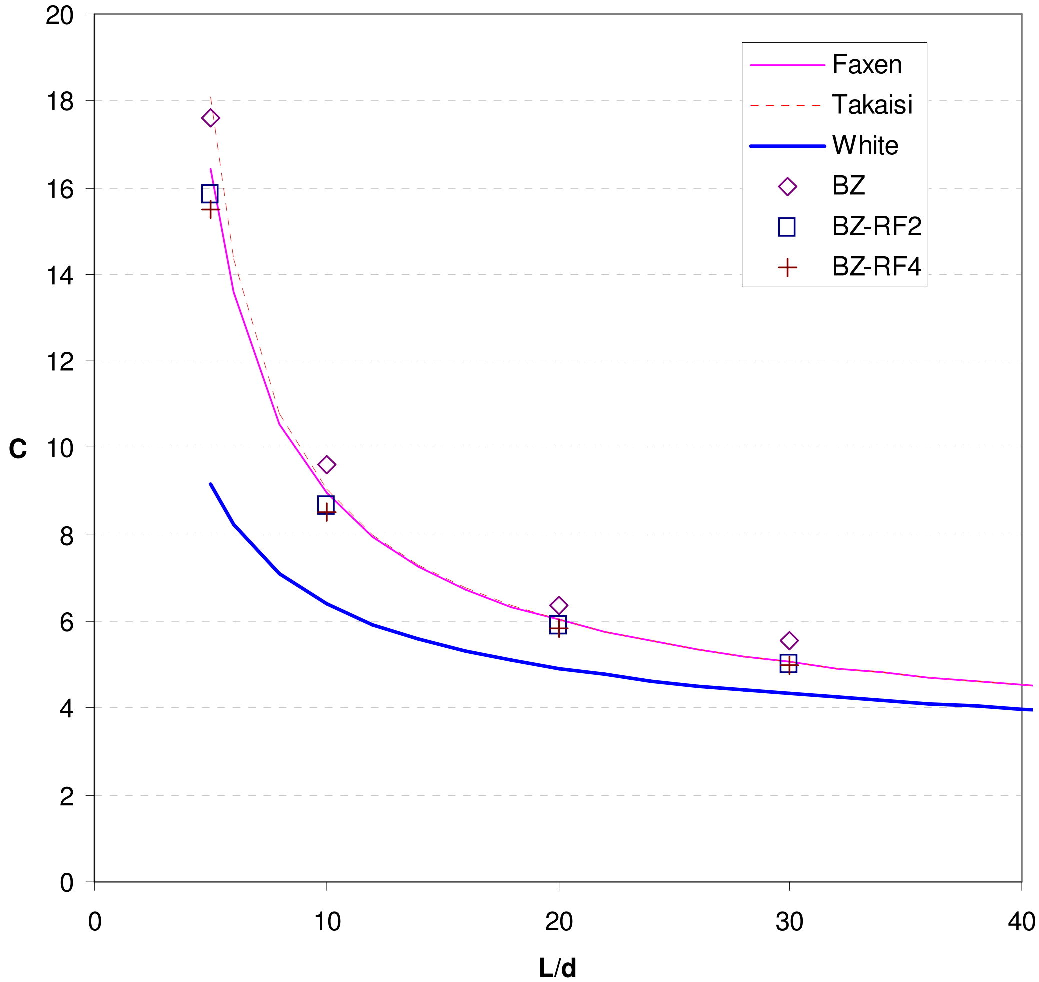


Figure C.64 – C versus L/d with Grid Refinement for the BZ Boundary Treatment

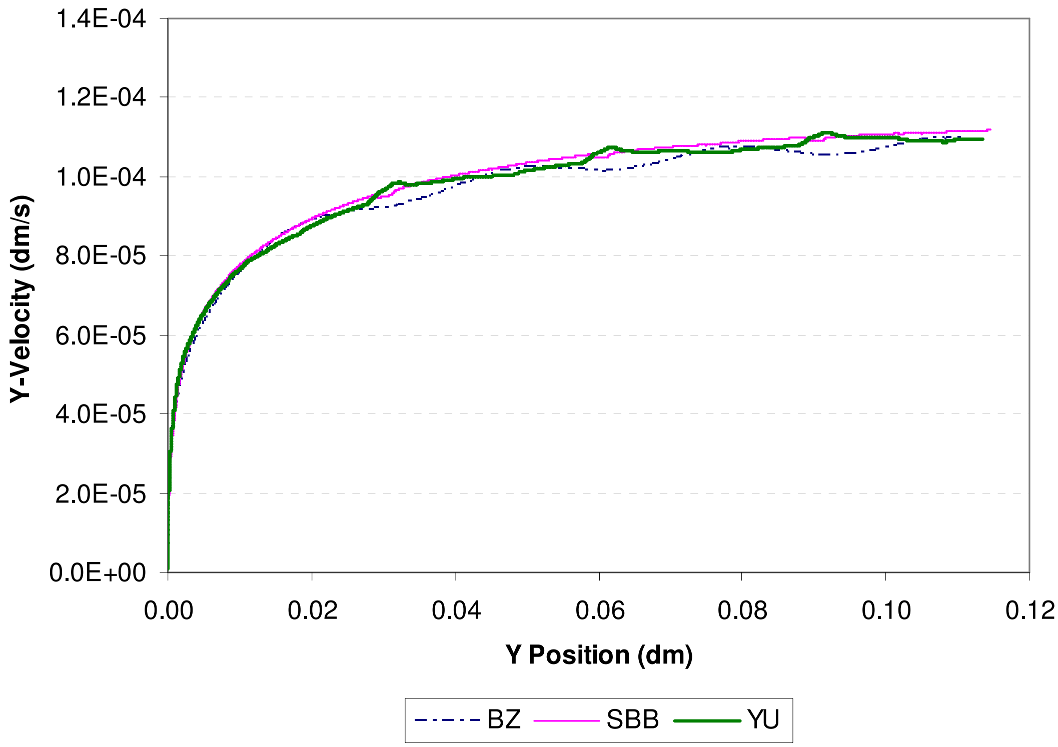


Figure C.65 – Particle Velocity versus Position for the VCF3 Grid

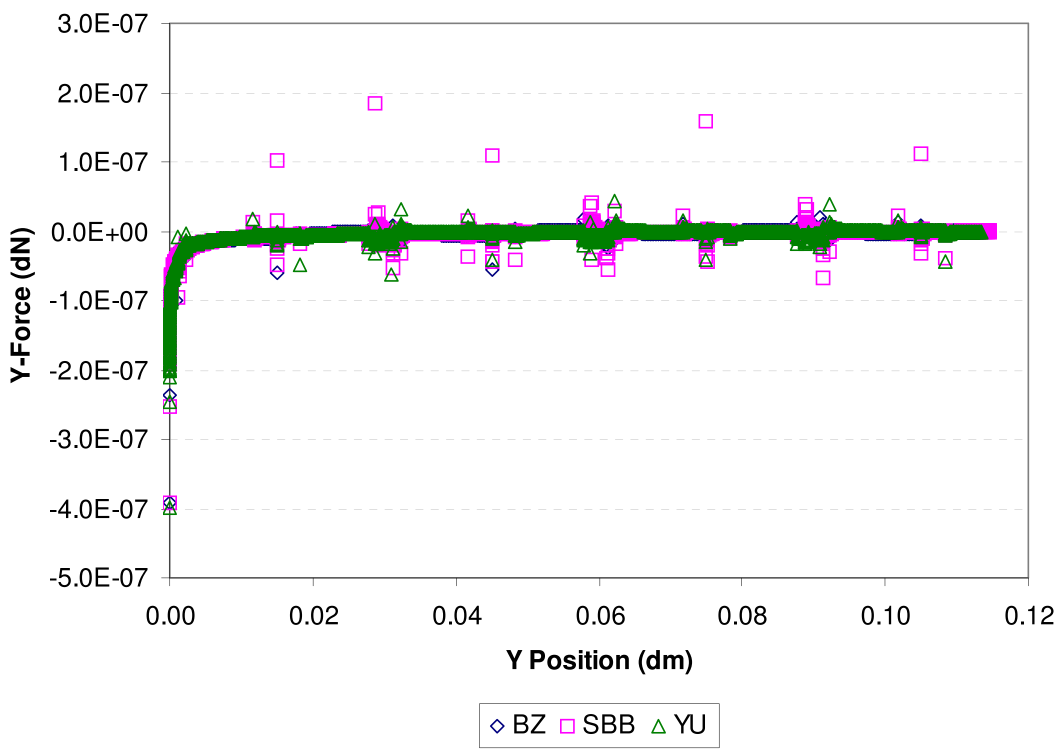


Figure C.66 – Particle Force versus Position for the VCF3 Grid

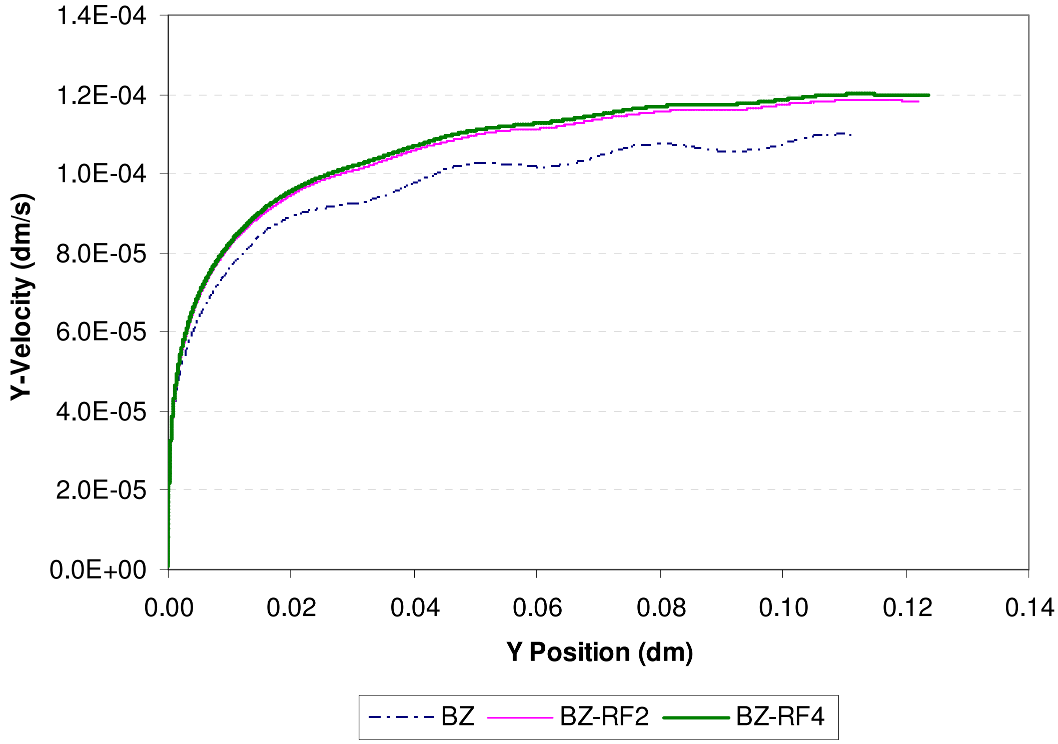


Figure C.67 – Particle Velocity versus Position for the VCF3-RF Grids

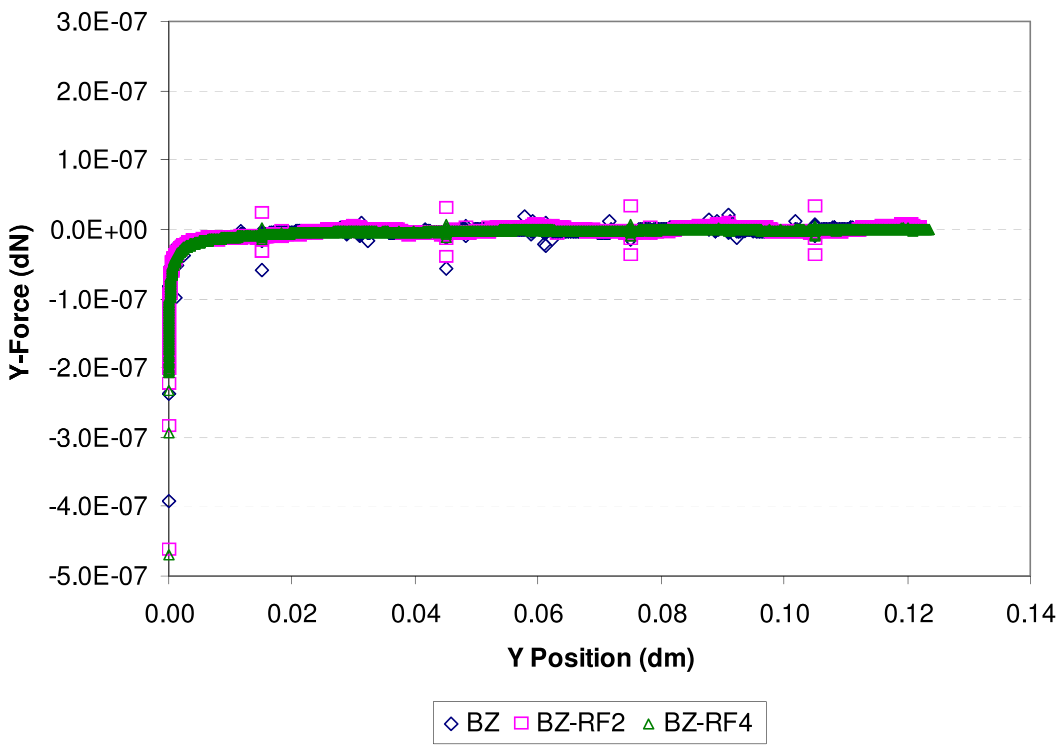


Figure C.68 – Particle Force versus Position for the VCF3-RF Grid

Two Particle Sedimentation in a Channel

Particle Path Plots

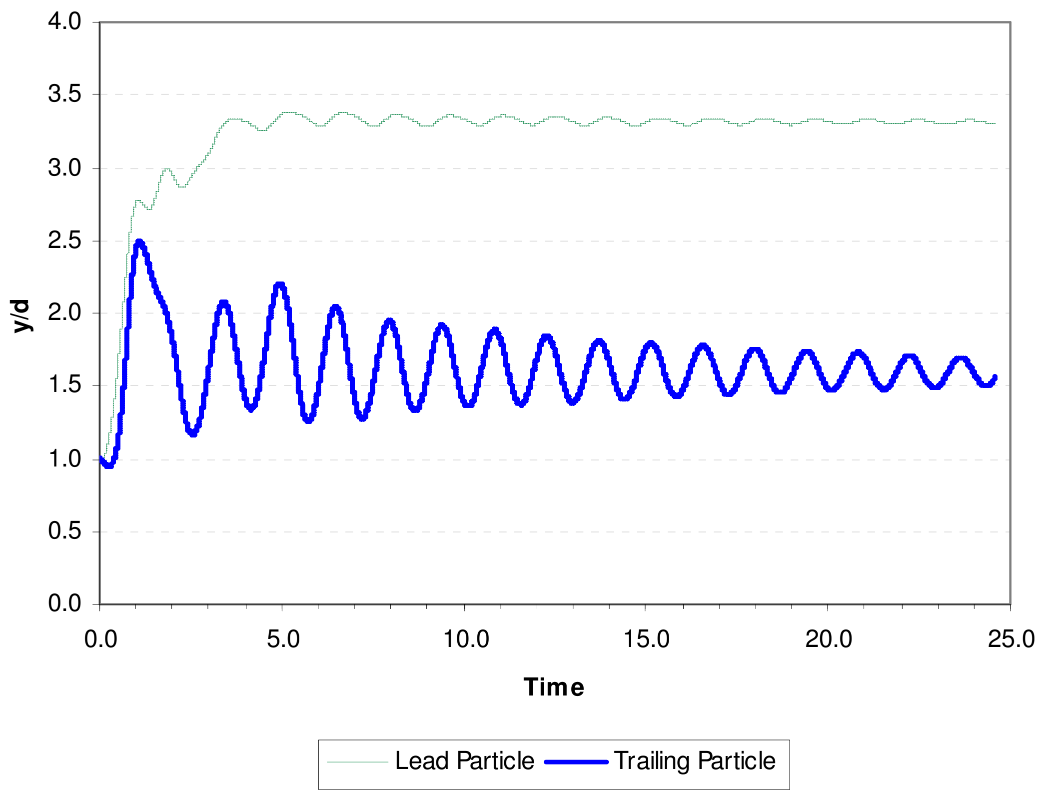


Figure C.69 – Typical Result for Two Particle Sedimentation in a Channel

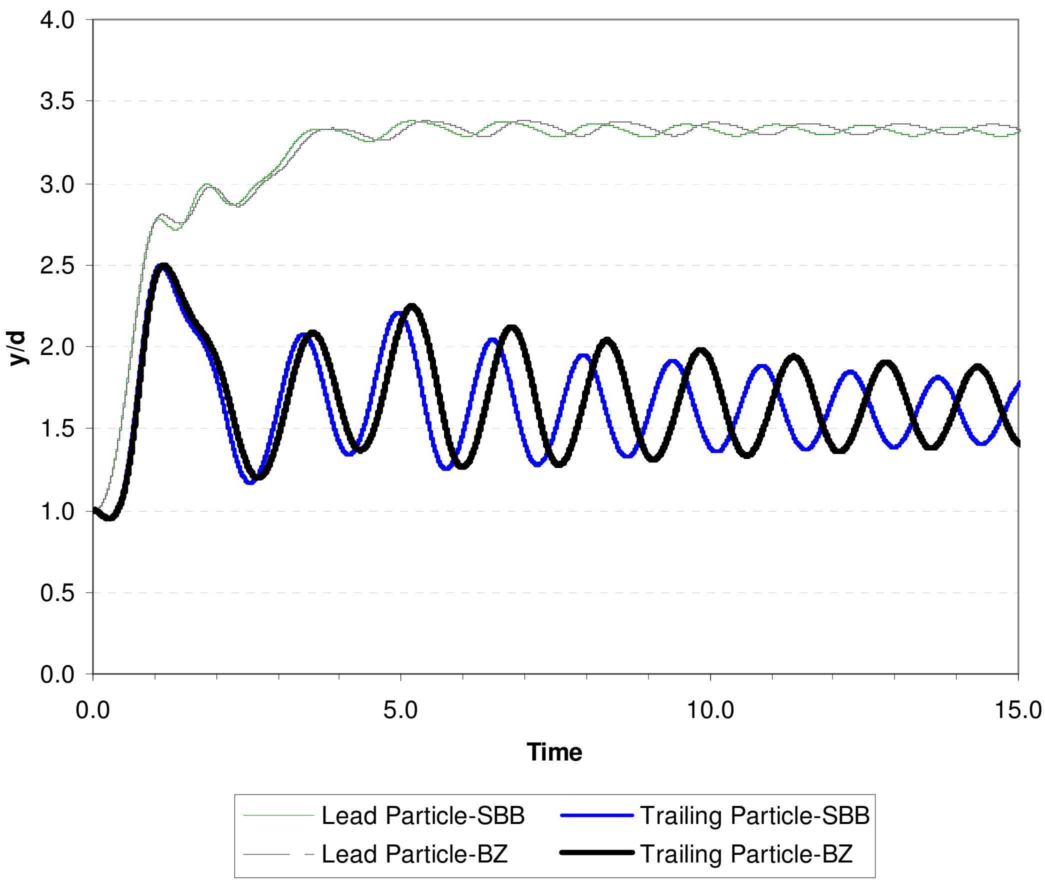


Figure C.70 – Two Particle Sedimentation Comparison SBB versus BZ

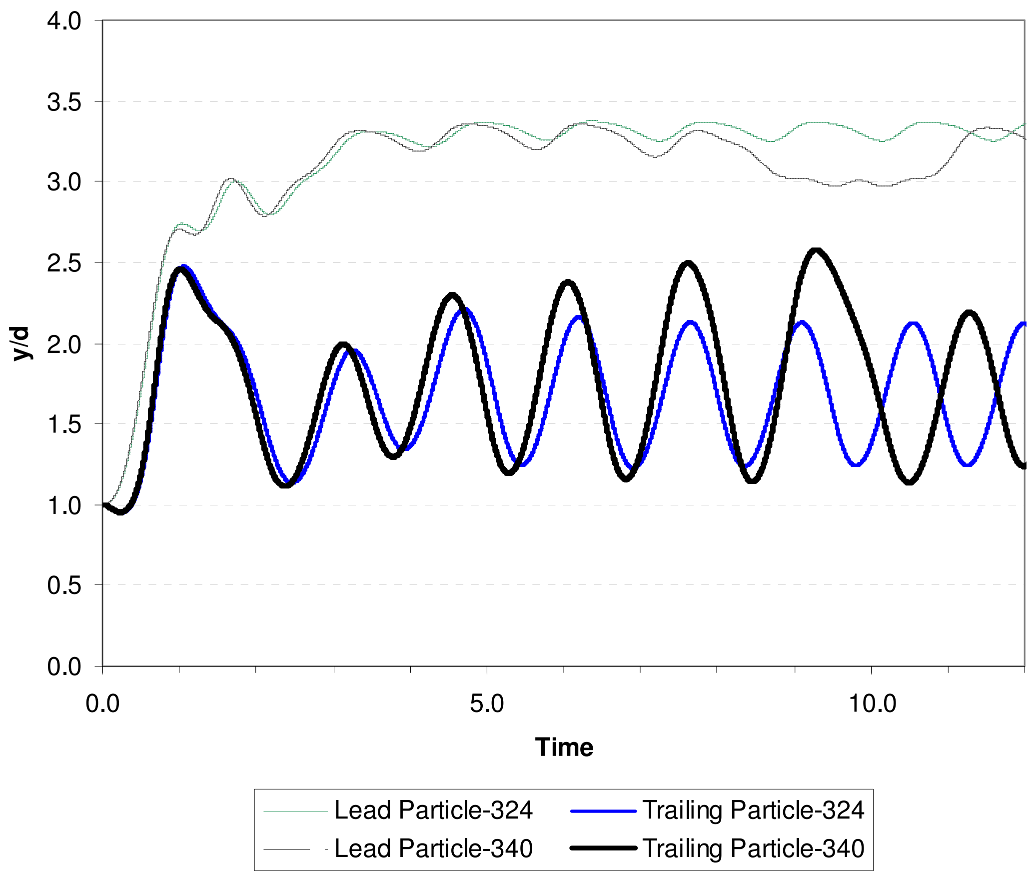


Figure C.71 – Two Particle Sedimentation Comparison for Net Body Forces

Particle Path Plots

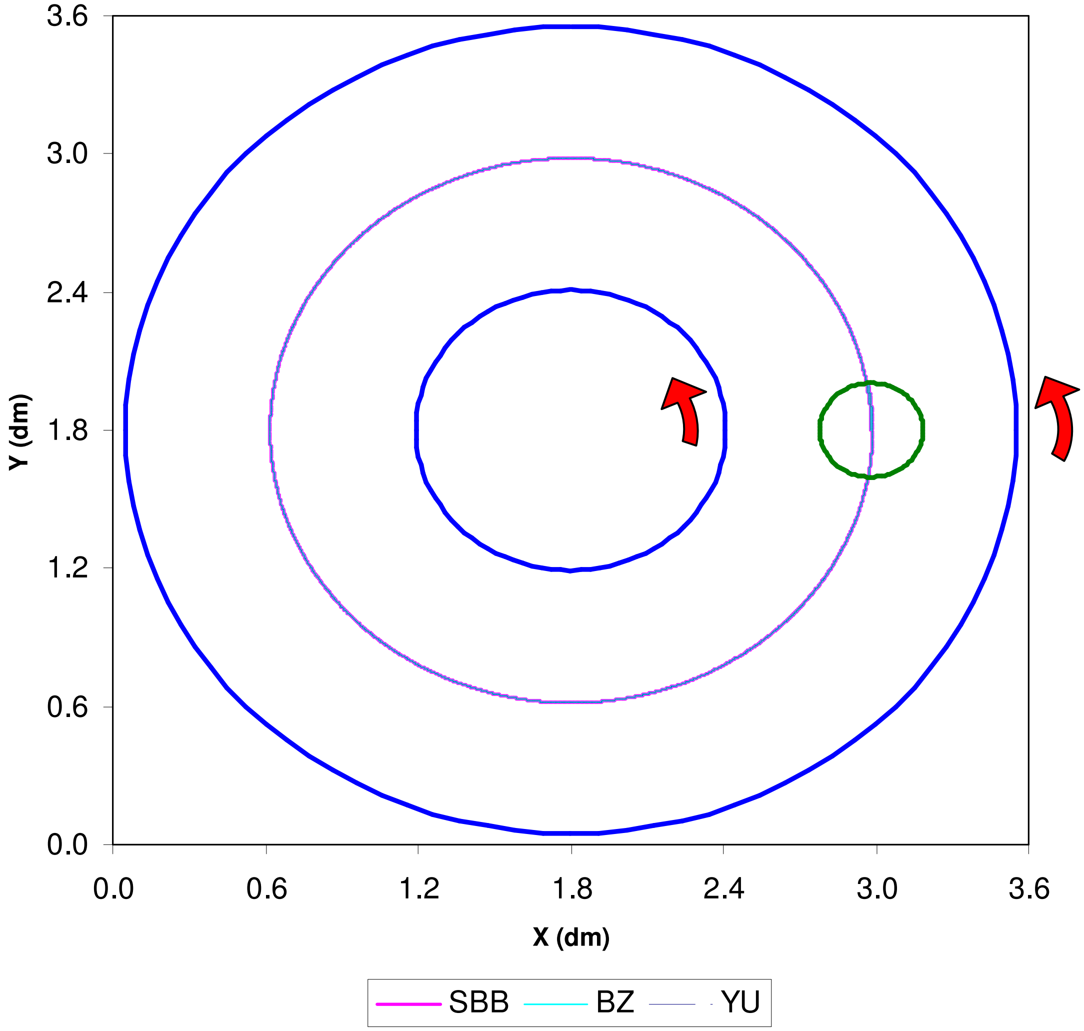


Figure C.72 – Circle Particle Path for ANN3 Case

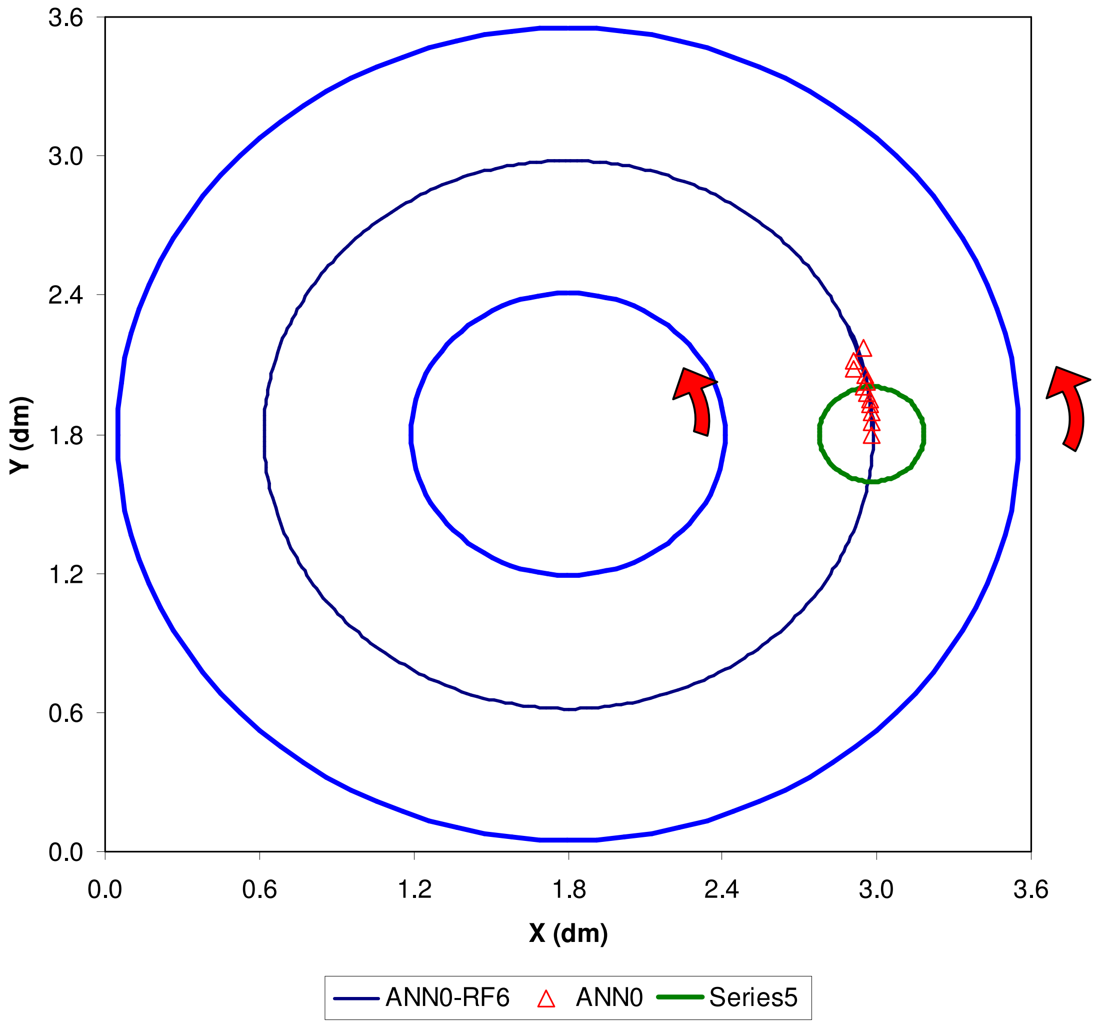


Figure C.73 – Circle Particle Path for ANN0-RF6 Case

Velocity Vector Visualization

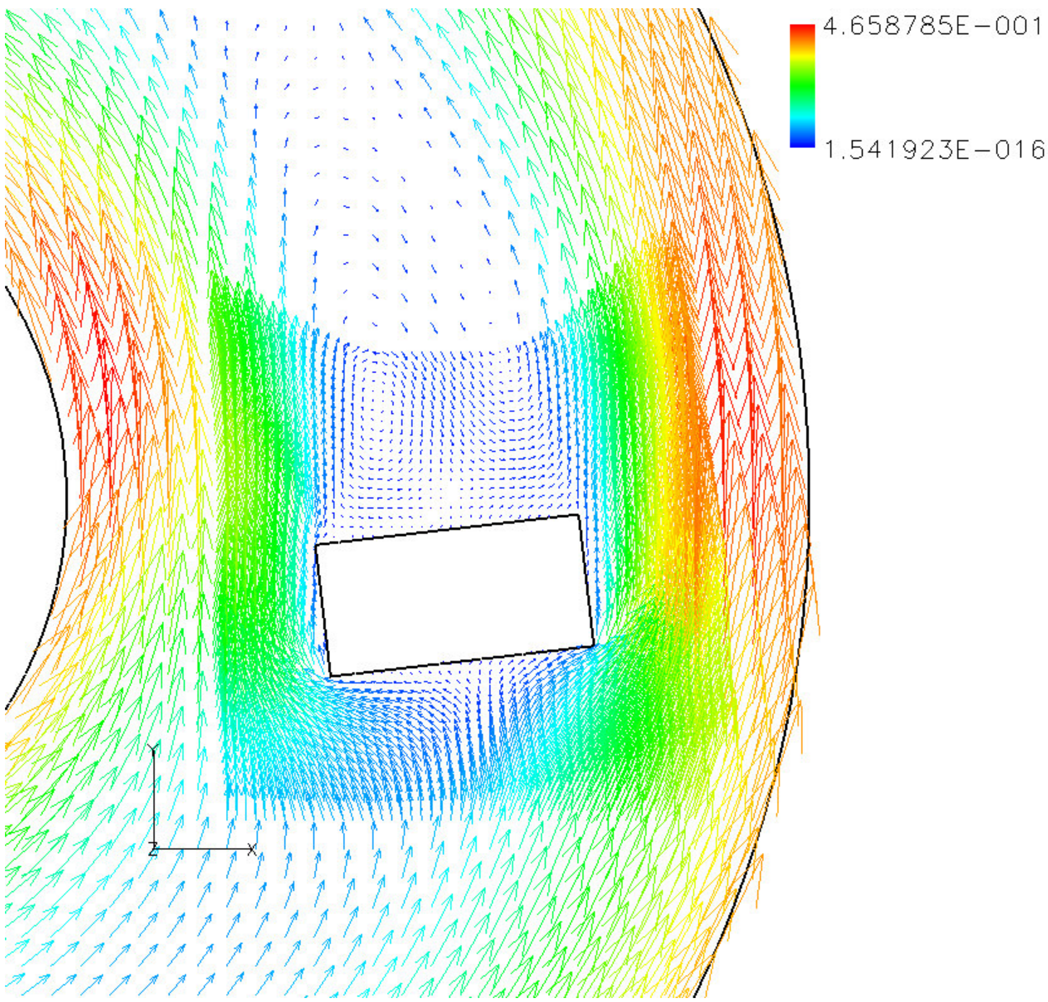


Figure C.74 – 2D Velocity Vectors for Flow around a RECT Construct

Construct Motion Paths

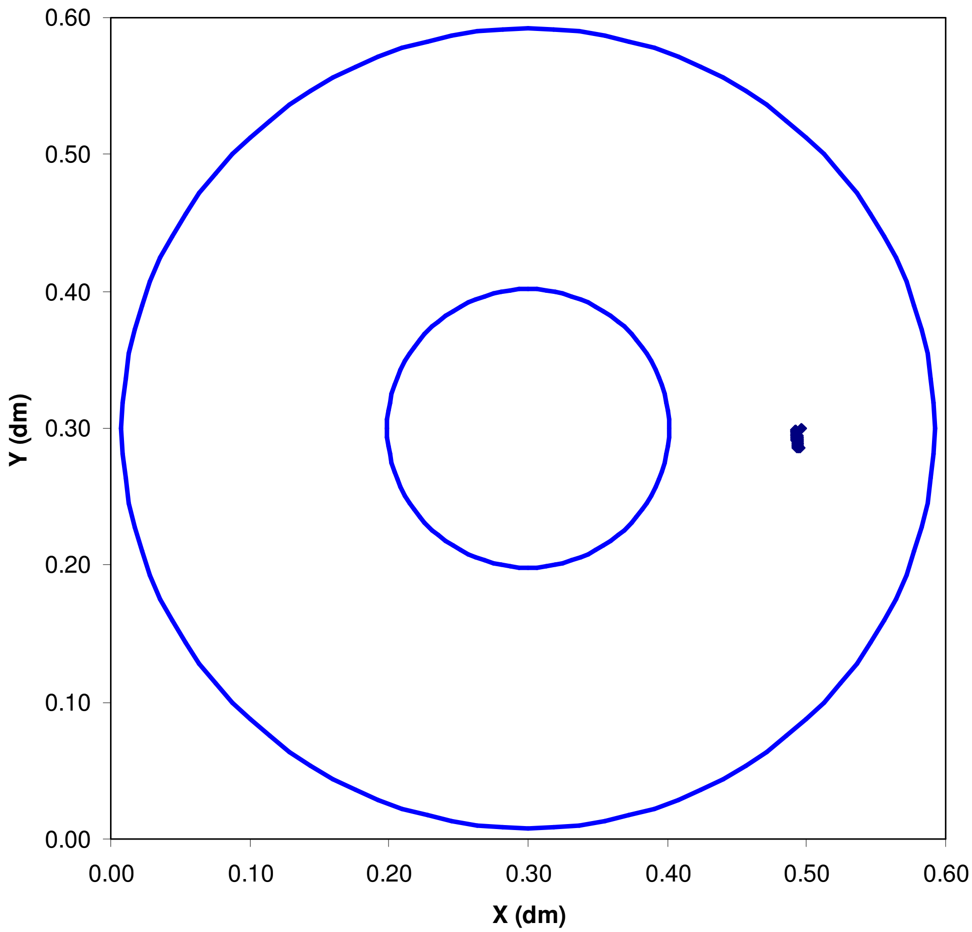


Figure C.75 – RECT Construct Path for Stationary Case

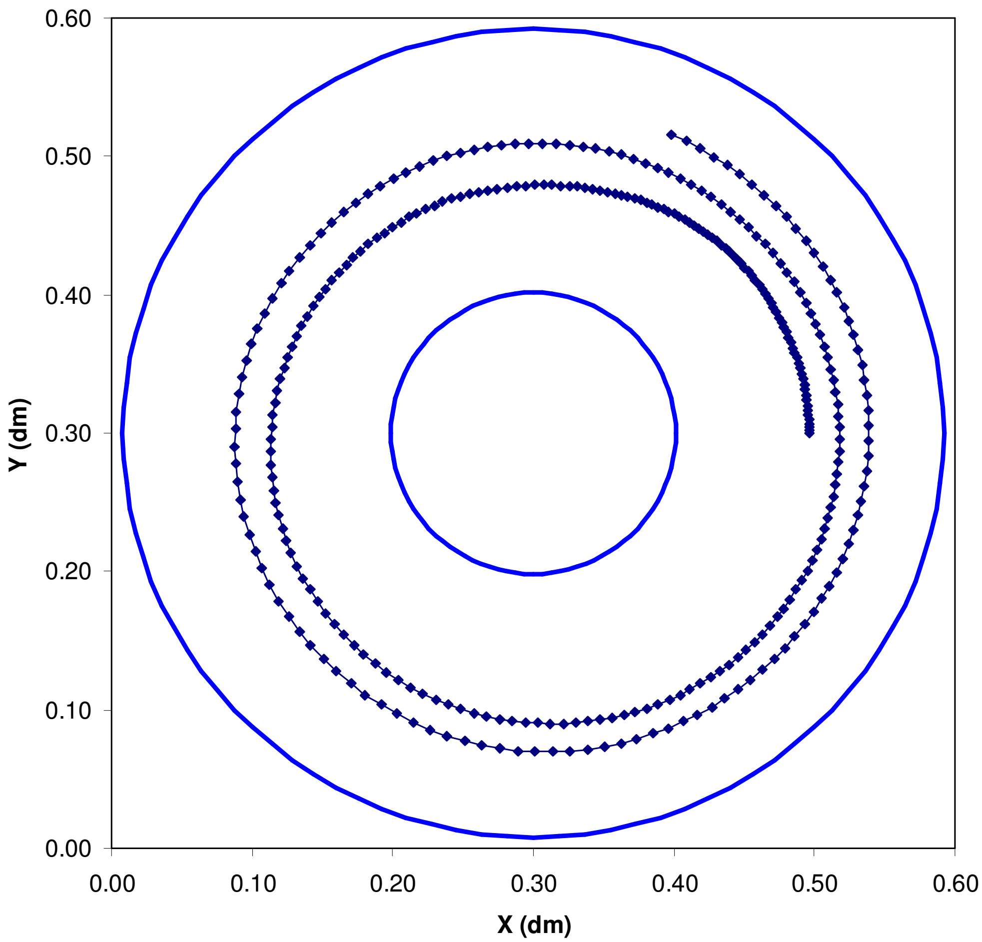


Figure C.76 – RECT Construct Path for Orbiting Case

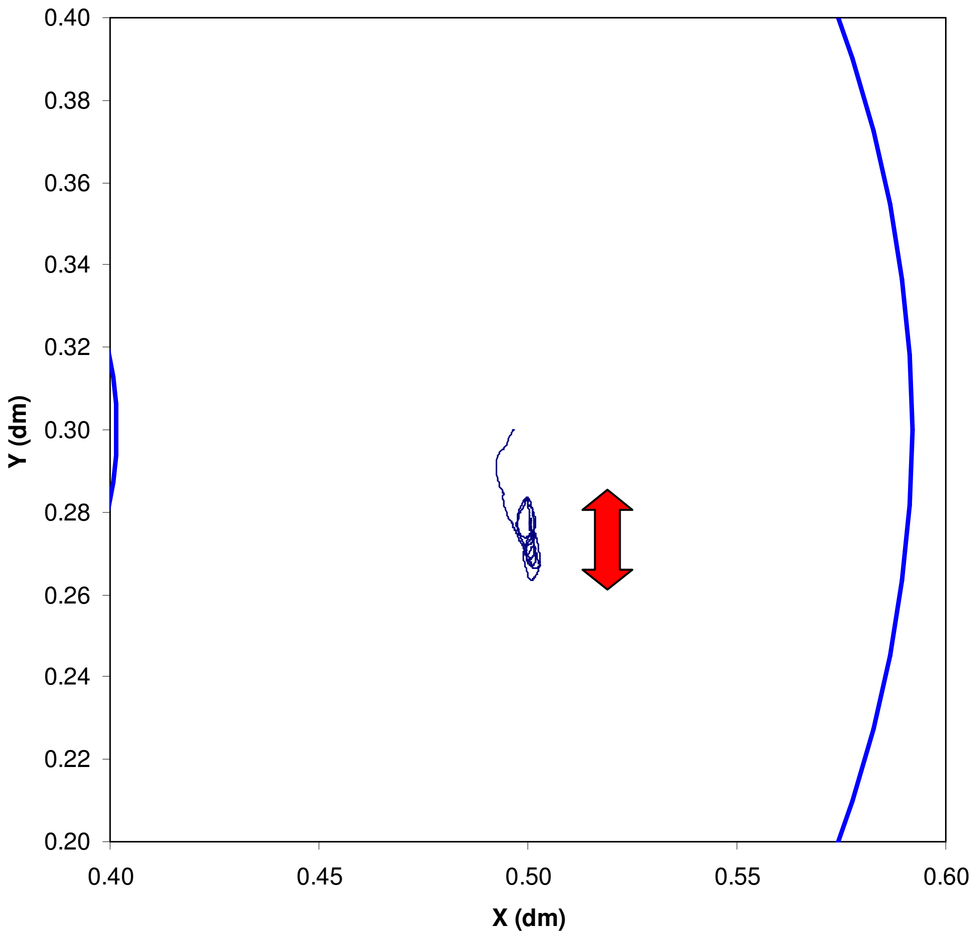


Figure C.77 – RECT Construct Path for Oscillation Case

Shear Stress Contour Plots

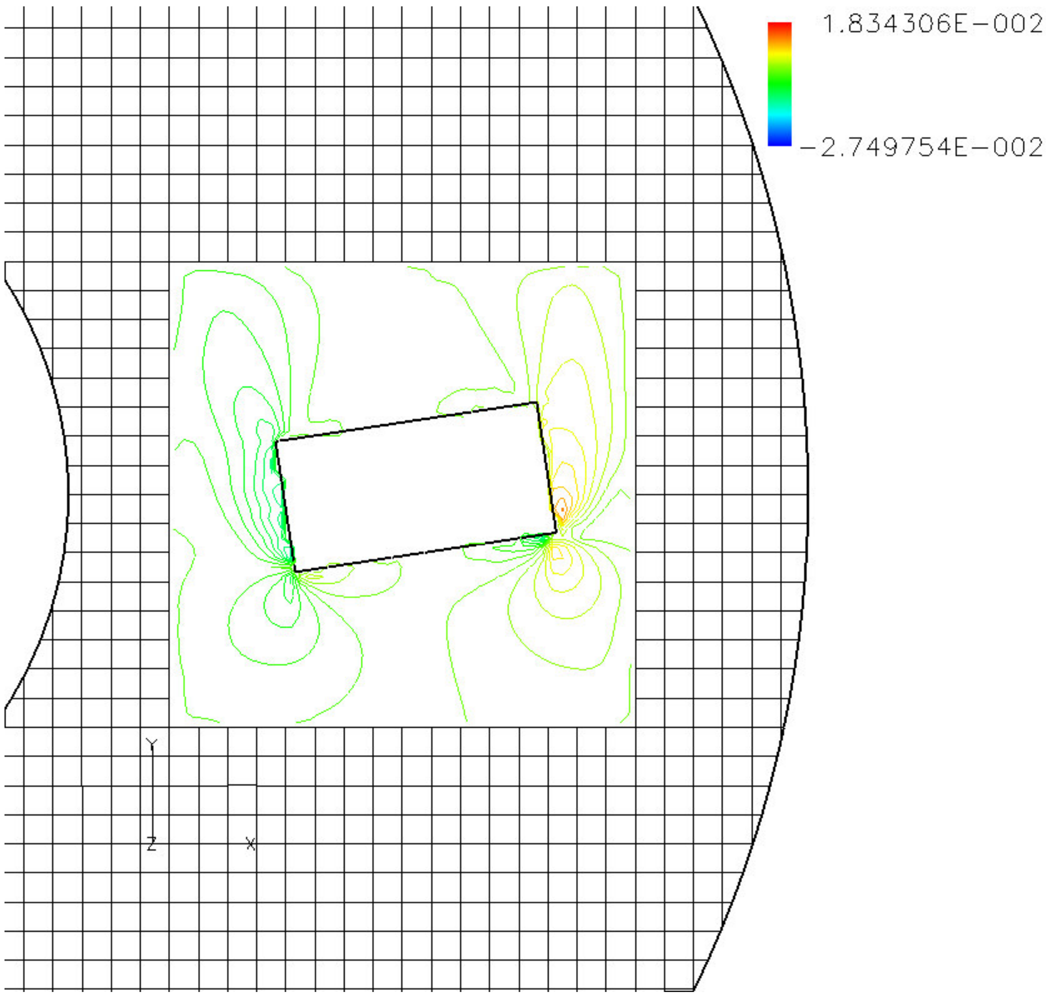


Figure C.78 – XY Shear Stress Contours for the RECT Construct

Appendix D

3D Results

The majority of the tables and figures that comprise the results from the 3D numerical studies that are not included in the body of this thesis are listed in this appendix. The intent is to provide a more complete record of the data gathered for this study. The tables and figures are referenced in the 3D results section of Chapter 4. This appendix is organized by numerical study.

Poiseuille Flow with shifted boundaries

<u>RMS Error plots</u>

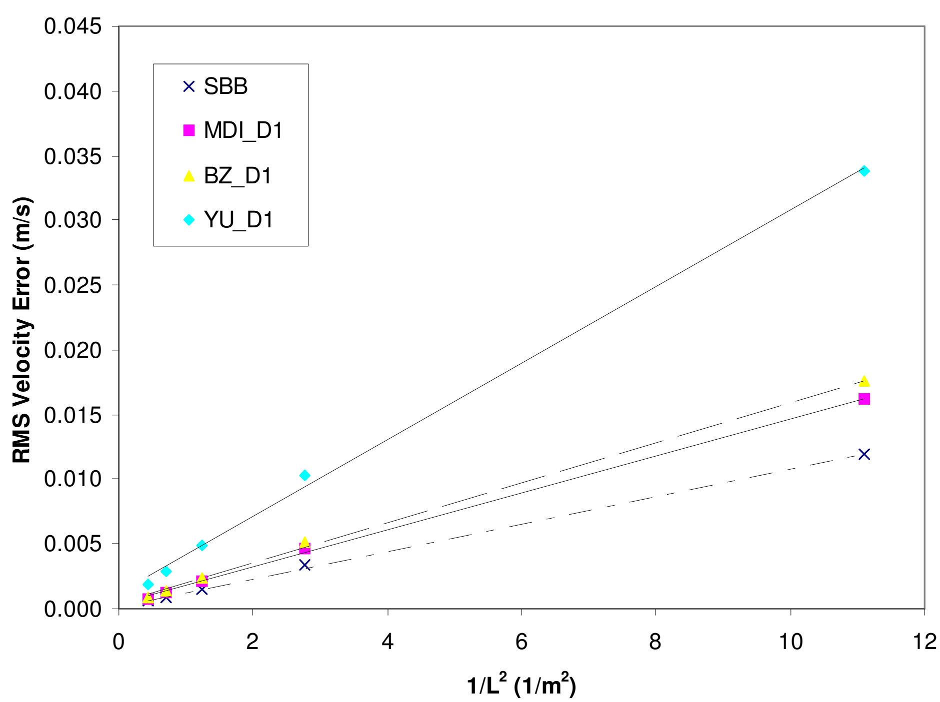


Figure D.1 – RMS Error versus $1/L^2$ for the 3D D1 Case

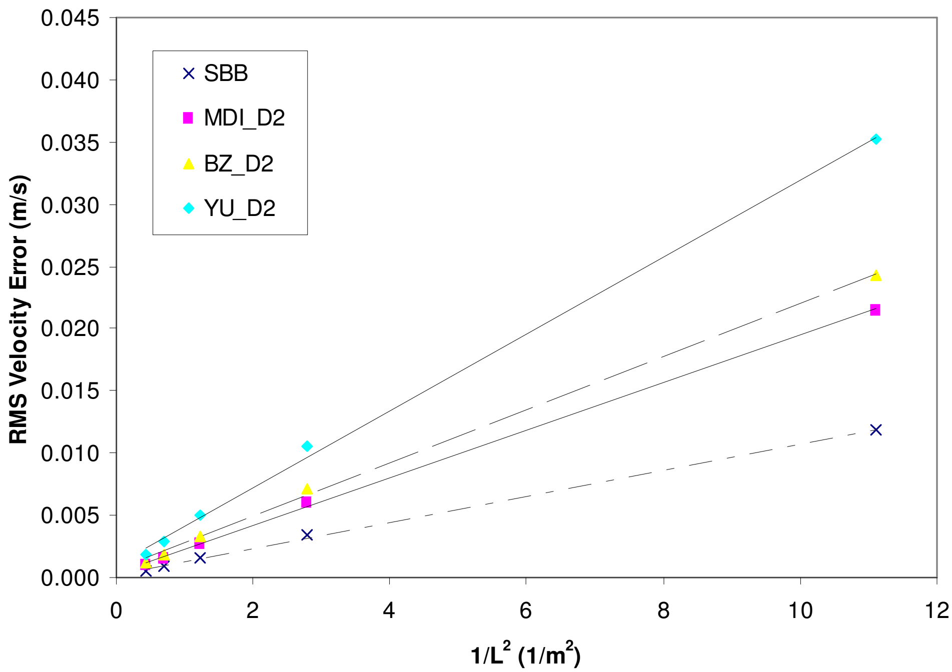


Figure D.2 – RMS Error versus $1/L^2$ for the 3D D2 Case

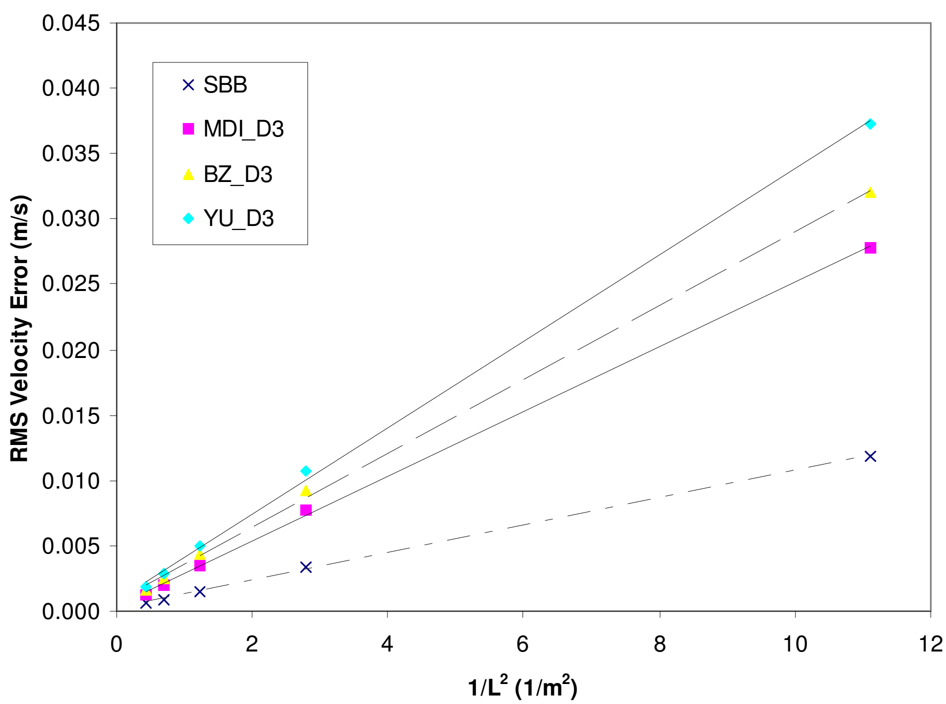


Figure D.3 – RMS Error versus $1/L^2$ for the 3D D3 Case

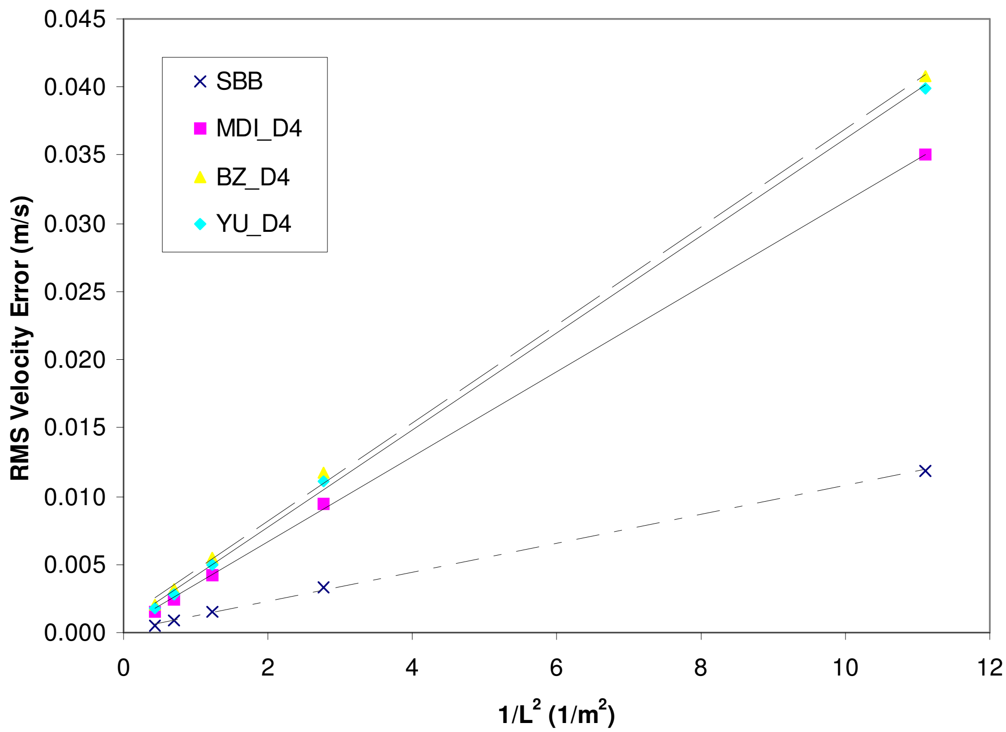


Figure D.4 – RMS Error versus $1/L^2$ for the 3D D4 Case

<u>RMS Error Results for HCF Cases</u>

Table D.1 – Channel Resolution (L) versus RMS Error for All Cases

		RMS Error (m/s)					
		D1 case			D2 case		
L (m)	SBB	MDI	BZ	YU	MDI	BZ	YU
0.3	0.01189	0.01617	0.01757	0.03385	0.02153	0.02428	0.03520
0.6	0.00337	0.00465	0.00512	0.01035	0.00610	0.00709	0.01051
0.9	0.00155	0.00214	0.00237	0.00496	0.00279	0.00330	0.00497
1.2	0.00088	0.00122	0.00136	0.00289	0.00159	0.00190	0.00288
1.5	0.00057	0.00079	0.00088	0.00189	0.00102	0.00123	0.00188
		D3 case			D4 case		
L (m)	SBB	MDI	BZ	YU	MDI	BZ	YU
0.3	0.01189	0.02784	0.03200	0.03730	0.03502	0.04074	0.03996
0.6	0.00337	0.00772	0.00927	0.01076	0.00949	0.01169	0.01107
0.9	0.00155	0.00349	0.00434	0.00499	0.00424	0.00548	0.00502
1.2	0.00088	0.00197	0.00251	0.00286	0.00238	0.00318	0.00284
1.5	0.00057	0.00126	0.00163	0.00185	0.00151	0.00207	0.00182

Streamline Visualizations for 3D LDCF Cases

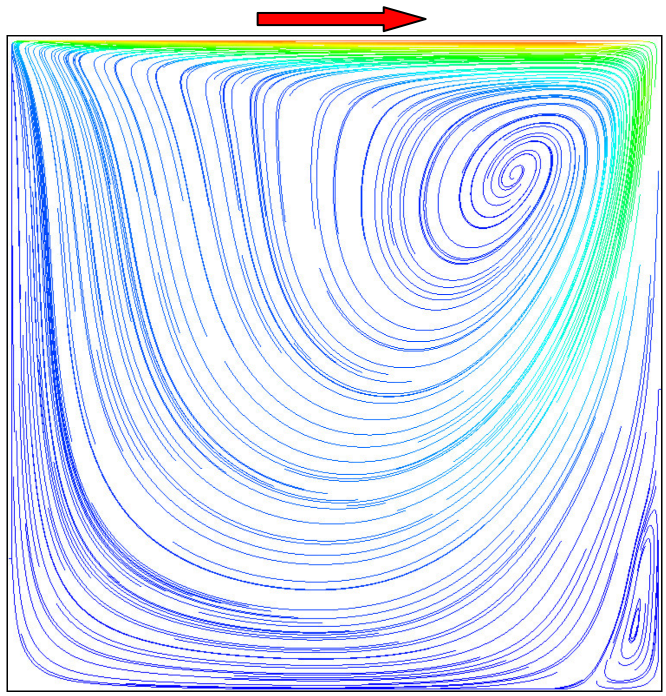


Figure D.5 – Streamlines for the 3D FLUENT LDCF at Z = 2.4dm

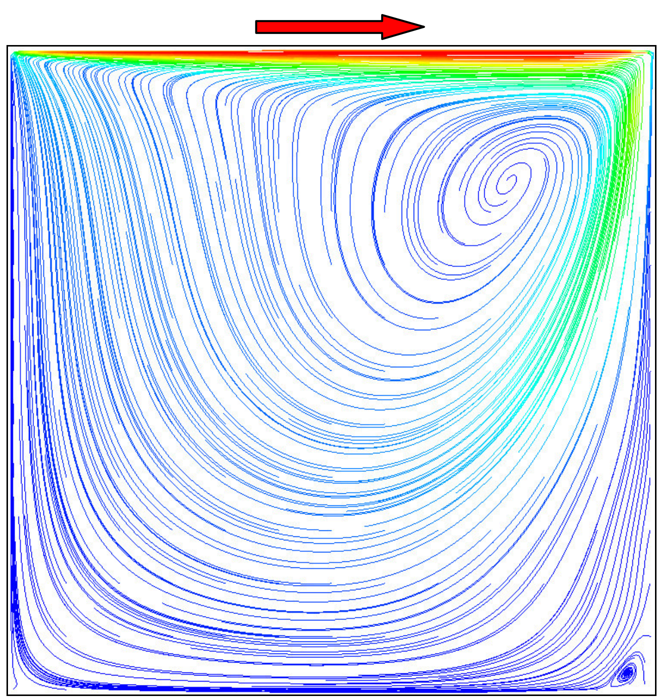


Figure D.6 – Streamlines for the 3D LB LDCF at Z = 2.4dm

Velocity Centerline Plots at Z = 1.2dm for 3D LDCF Cases

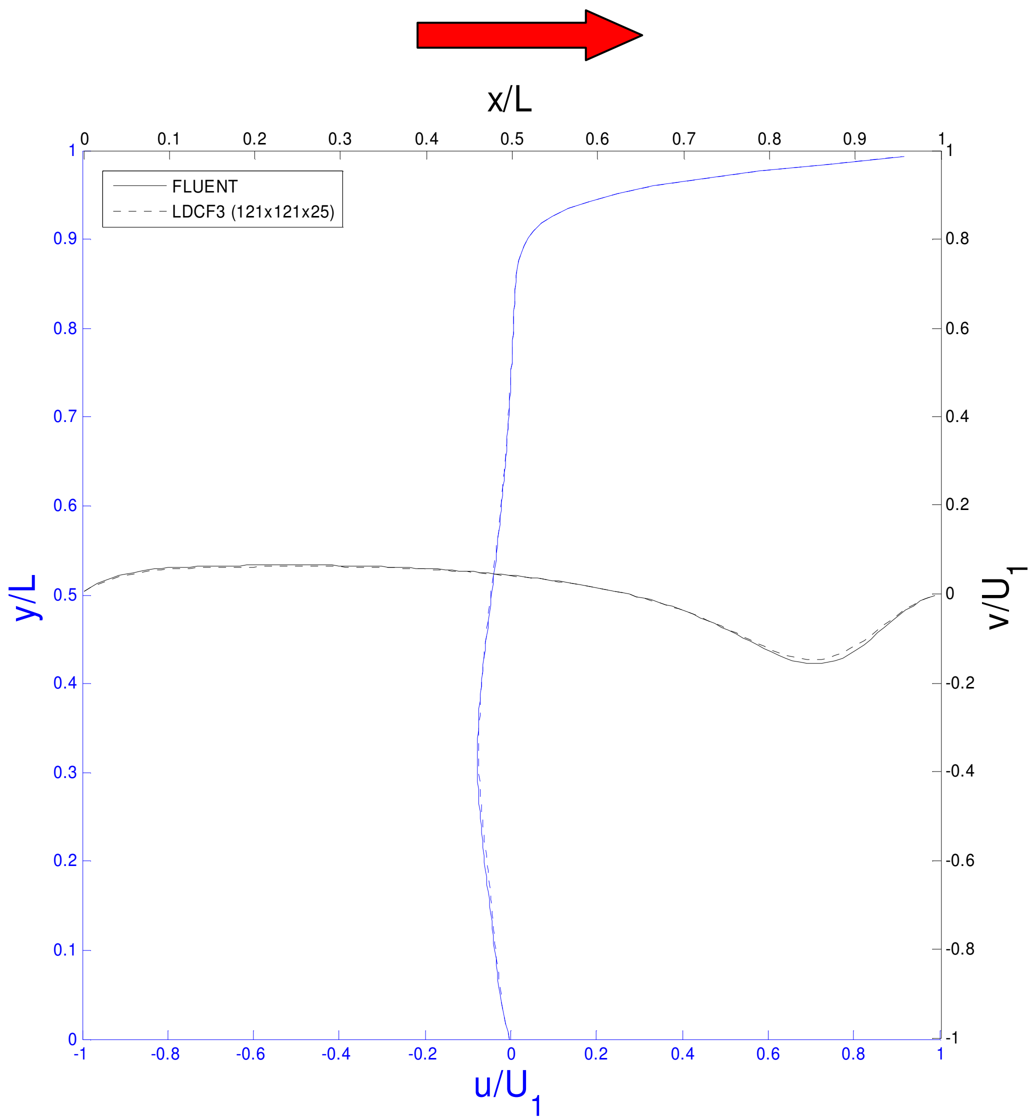


Figure D.7 – 3D LDCF Velocity Centerline Plots using the SBB Treatment

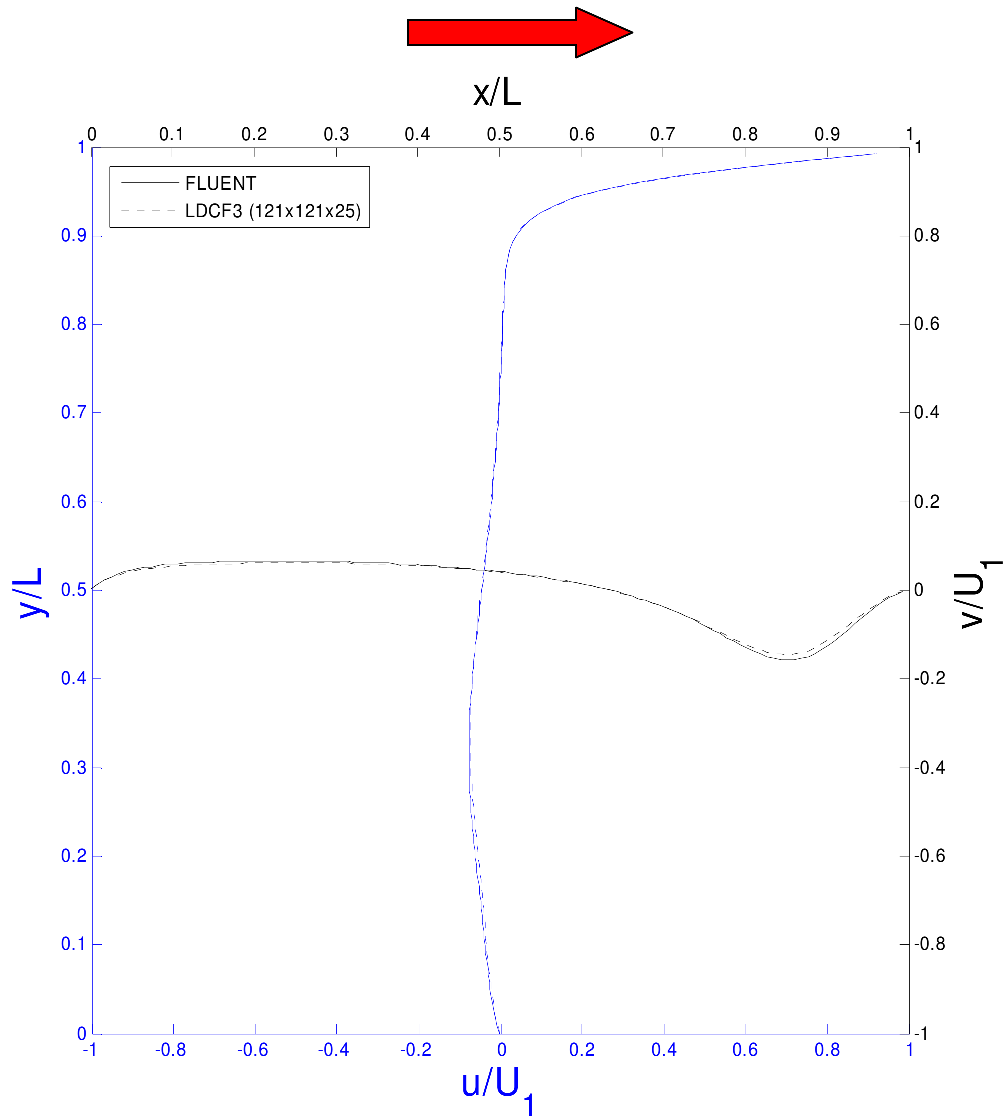


Figure D.8 – 3D LDCF Velocity Centerline Plots using the BZ Treatment

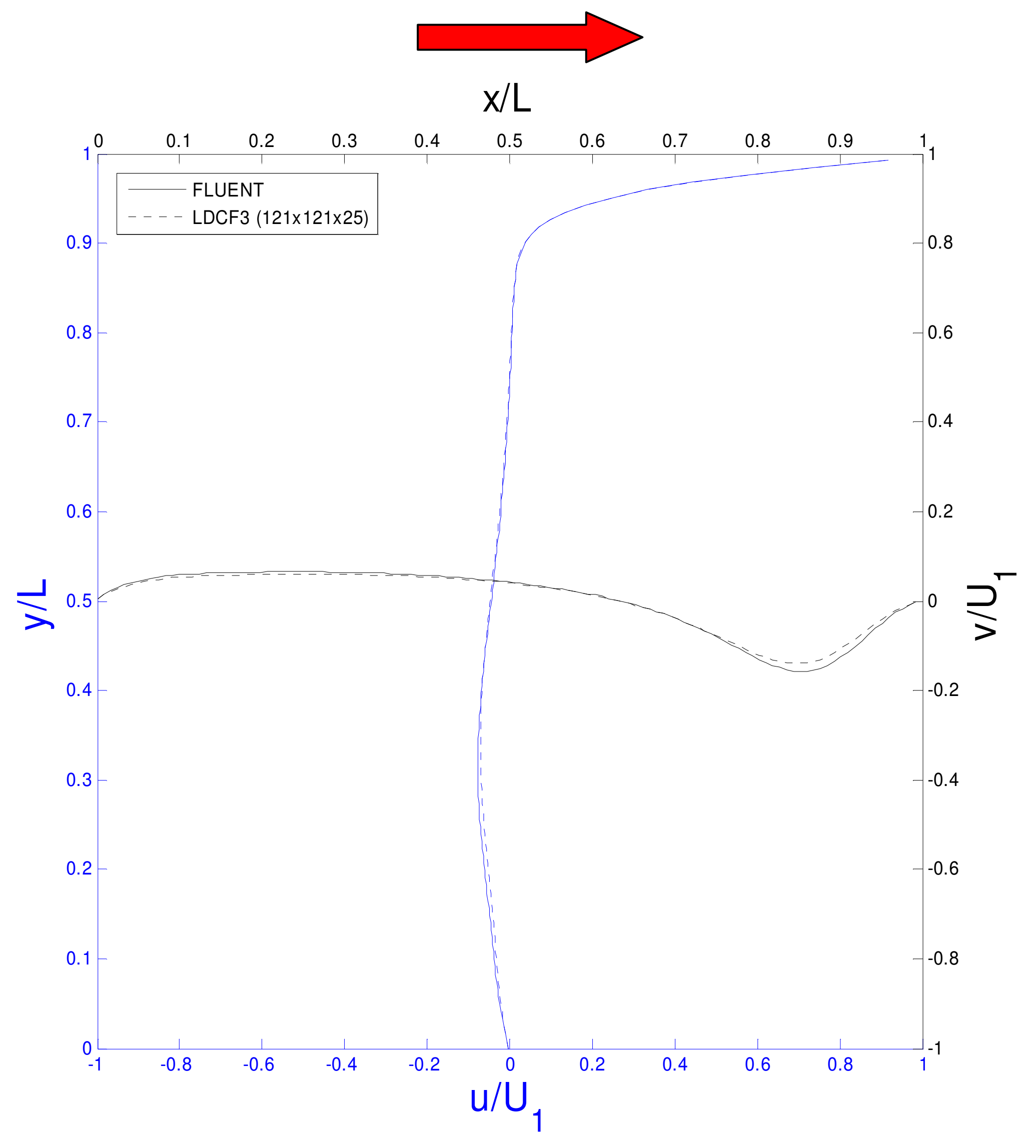


Figure D.9 – 3D LDCF Velocity Centerline Plots using the YU Treatment

Z-Velocity Plots at Z = 1.2dm, Y = 21dm for 3D LDCF Cases

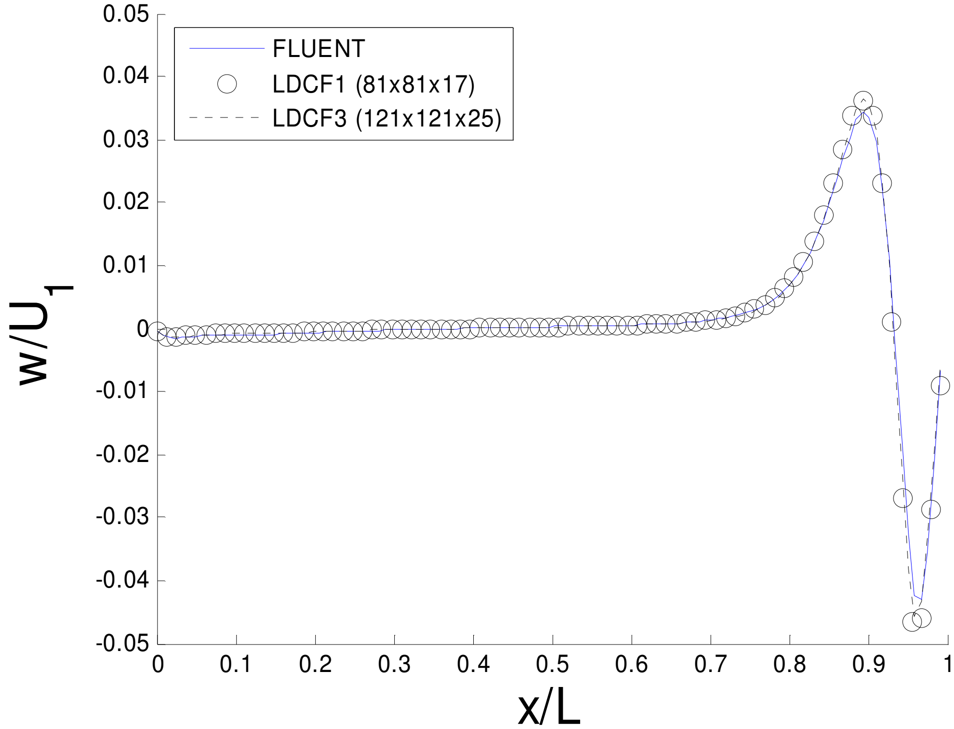


Figure D.10 – 3D LDCF Z-Velocity Plot using the SBB Treatment

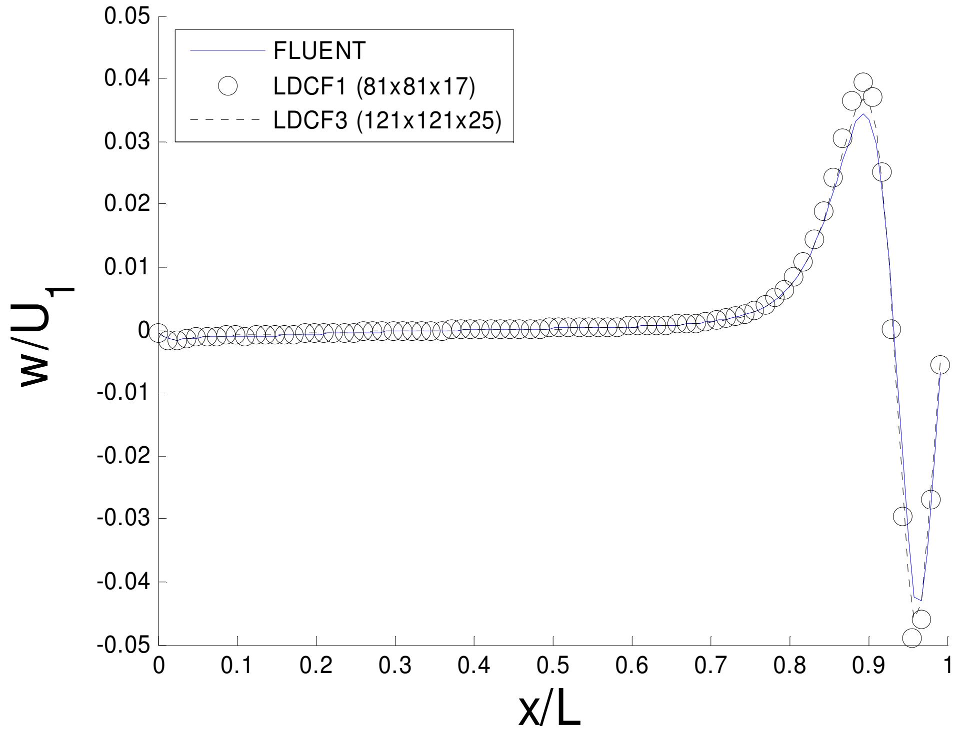


Figure D.11 – 3D LDCF Z-Velocity Plot using the BZ Treatment

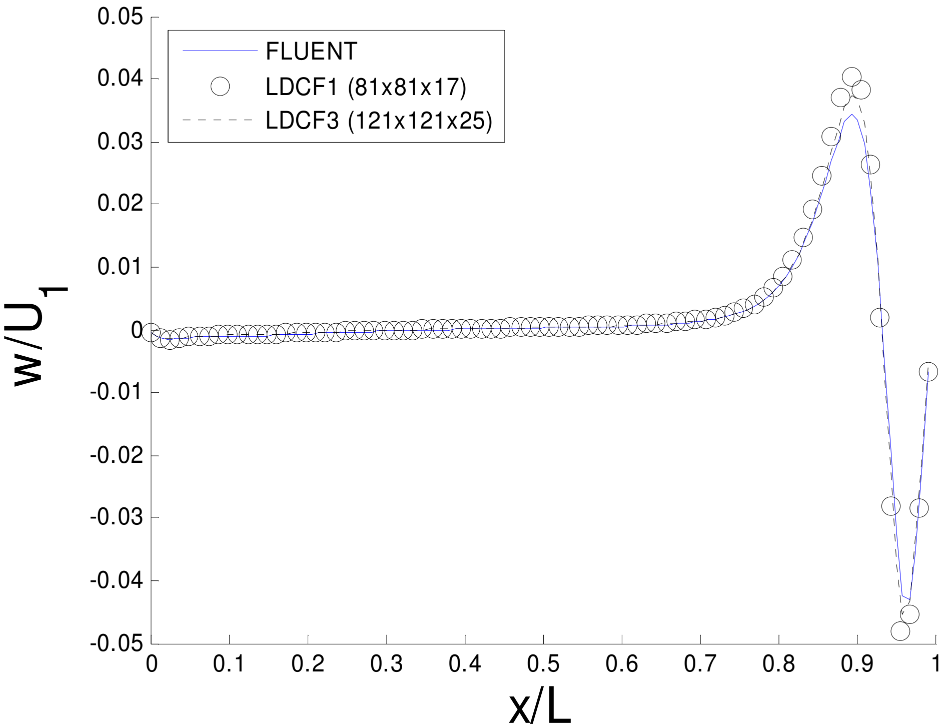


Figure D.12 – 3D LDCF Z-Velocity Plot using the YU Treatment

Absolute Error Visualizations for 3D LDCF Cases at Z = 2.4dm

Note: LDCF bottom left corner is at origin

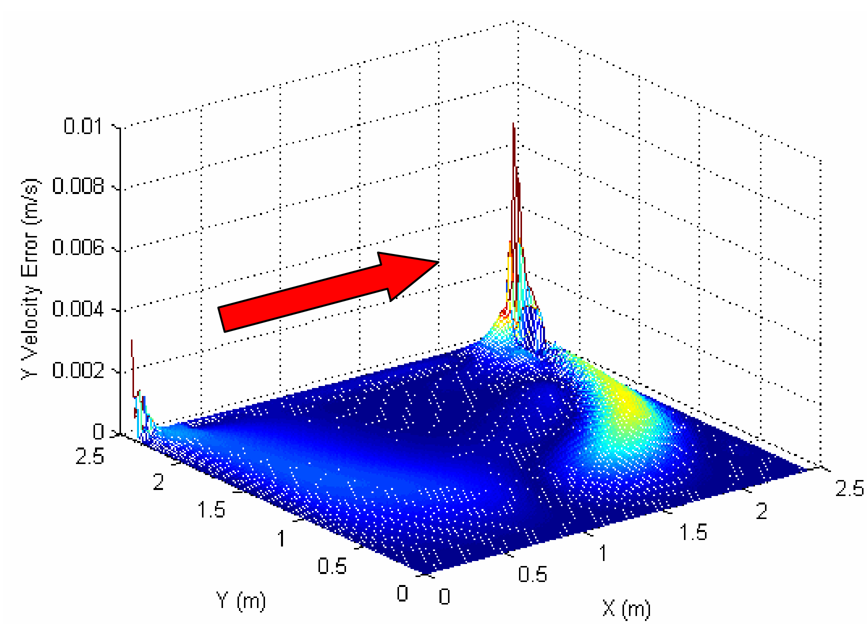


Figure D.13 – Y-Velocity Error for the LDCF3 case using the SBB Treatment

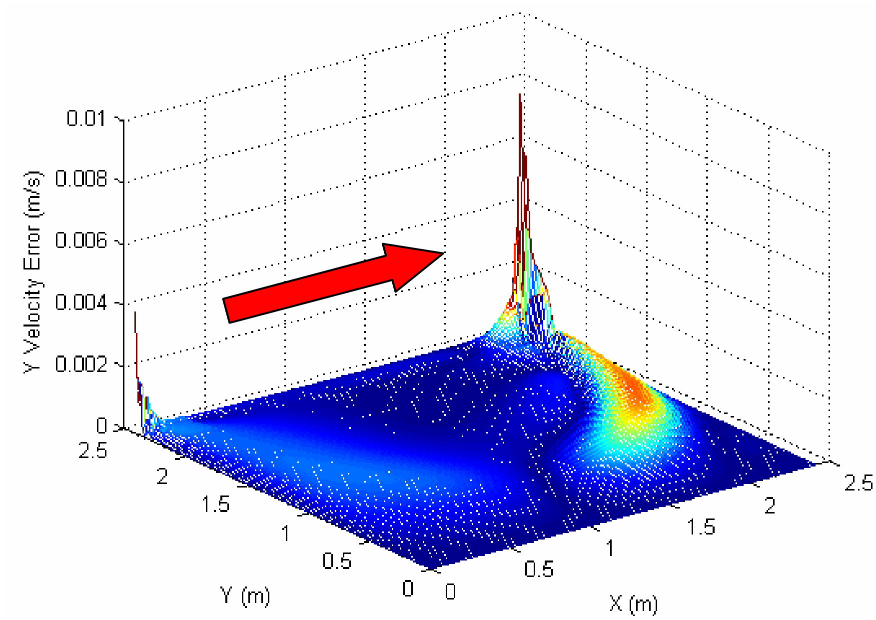


Figure D.14 – Y-Velocity Error for the LDCF3 case using the BZ Treatment

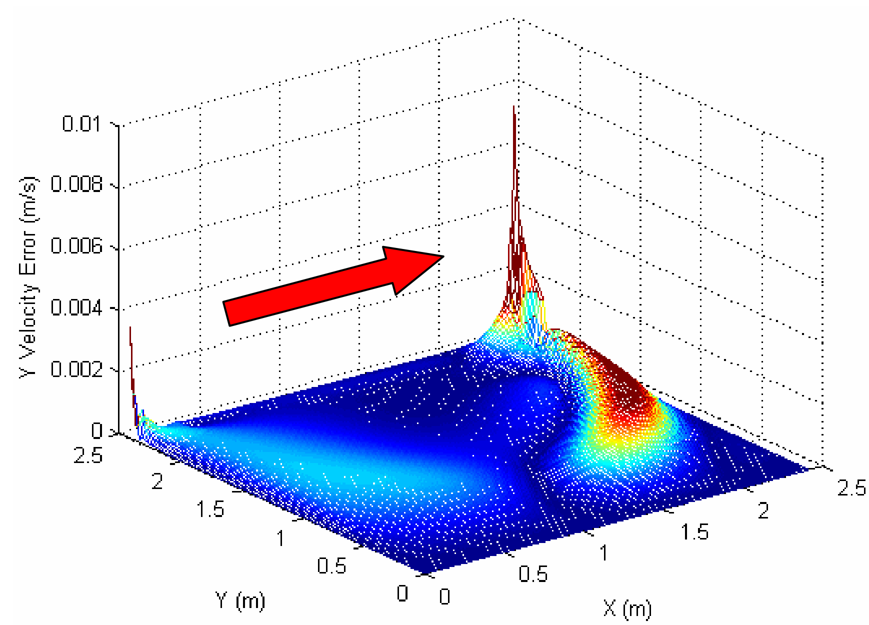


Figure D.15 – Y-Velocity Error for the LDCF3 case using the YU Treatment

RMS Error Plots for 3D LDCF Cases

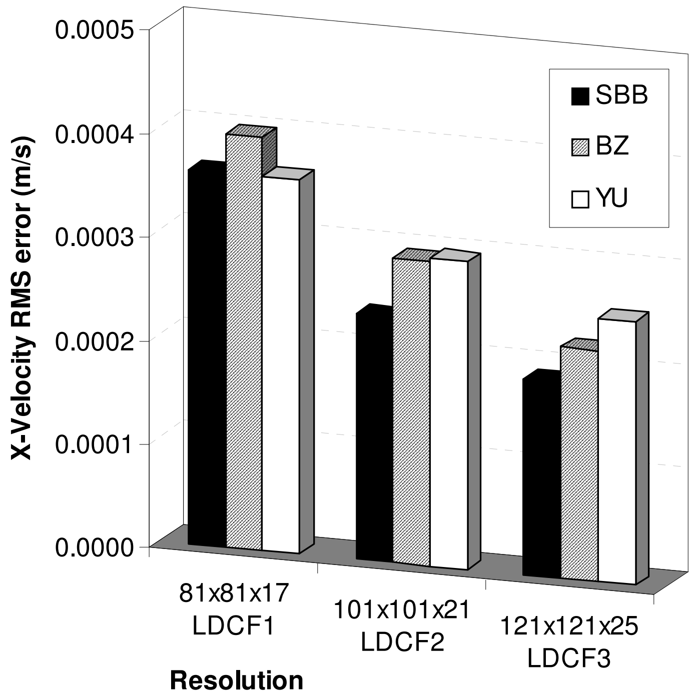


Figure D.16 – X-Velocity RMS Error for the 3D LDCF Cases at Z = 2.4dm

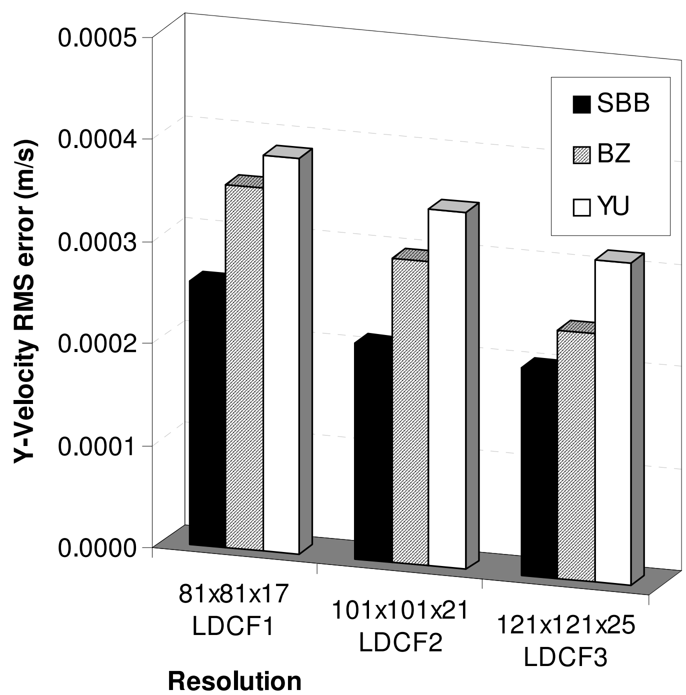


Figure D.17 – Y-Velocity RMS Error for the 3D LDCF Cases at Z = 2.4dm

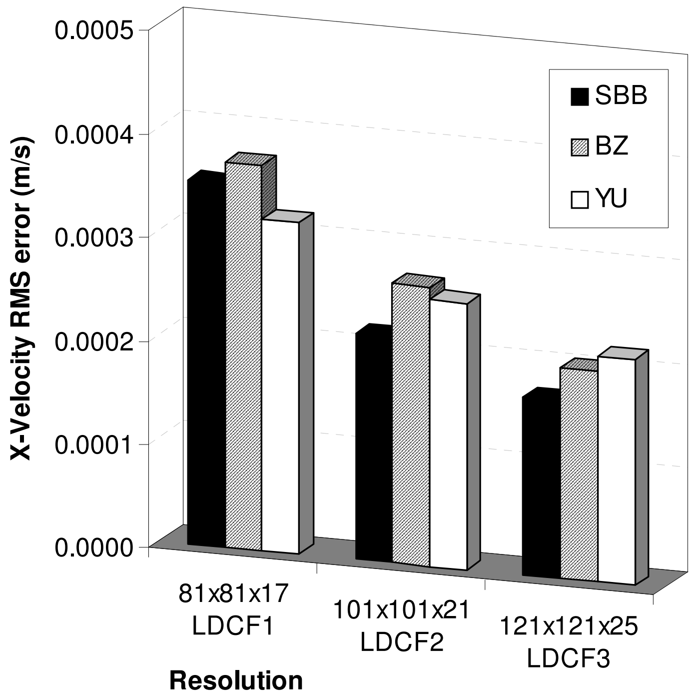


Figure D.18 – X-Velocity RMS Error for the 3D LDCF Cases at Z = 1.2dm

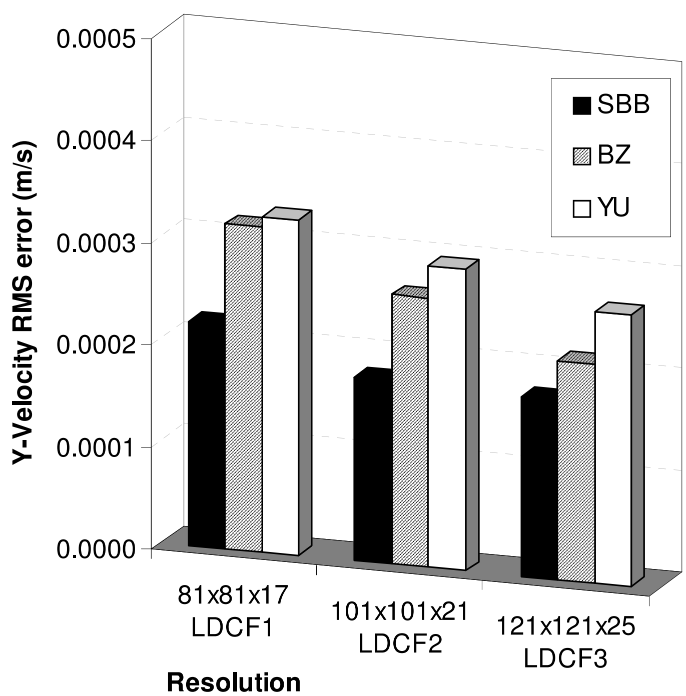


Figure D.19 – Y-Velocity RMS Error for the 3D LDCF Cases at Z = 1.2dm

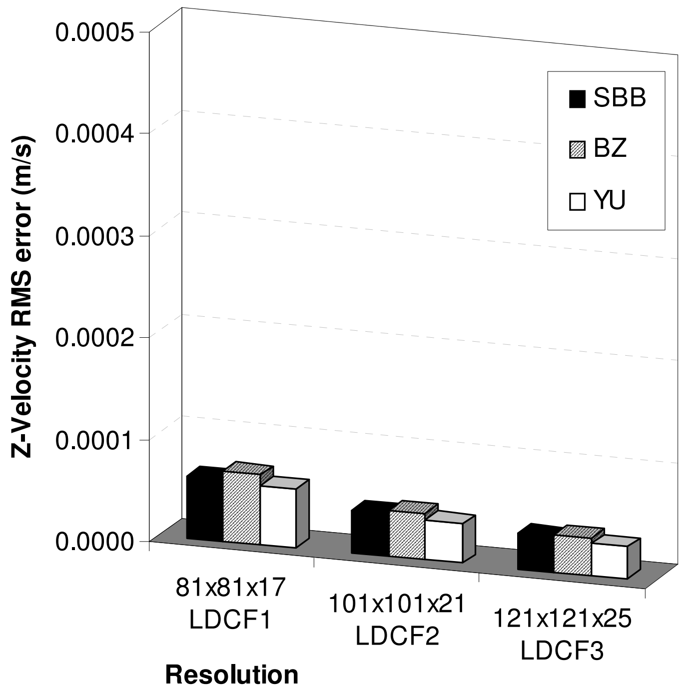


Figure D.20 – Z-Velocity RMS Error for the 3D LDCF Cases at Z = 1.2dm

Percent Error Results for 3D LDCF Cases

Table D.2 – X-Velocity Percent Error Results for the Z = 2.4 dm LDCF Cases

Case	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1 81x81x17	SBB	13.23	53.73	86.55
	BZ	7.94	30.75	53.52
	YU	6.14	27.59	44.25
LDCF2 101x101x21	SBB	11.18	53.76	90.32
	BZ	8.21	37.63	58.81
	YU	7.49	29.37	47.69
LDCF3 121x121x25	SBB	12.30	52.30	87.01
	BZ	11.32	42.95	71.69
	YU	9.41	30.99	50.50

Table D.3 – Y-Velocity Percent Error Results for the Z = 2.4 dm LDCF Cases

Case	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1 81x81x17	SBB	10.52	49.93	85.02
	BZ	5.07	30.27	55.03
	YU	4.51	23.15	45.48
LDCF2 101x101x21	SBB	12.11	49.17	83.20
	BZ	8.10	33.57	61.30
	YU	5.61	25.73	48.79
LDCF3 121x121x25	SBB	12.77	47.95	81.62
	BZ	10.77	40.08	71.37
	YU	6.63	28.52	52.39

Table D.4 – X-Velocity Percent Error Results for the Z = 1.2 dm LDCF Cases

Case	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1 81x81x17	SBB	14.89	61.43	86.55
	BZ	7.85	33.73	54.19
	YU	6.53	27.08	44.50
LDCF2 101x101x21	SBB	13.31	56.69	92.17
	BZ	9.31	38.14	59.78
	YU	8.31	28.68	47.89
LDCF3 121x121x25	SBB	13.71	51.62	87.74
	BZ	12.36	43.28	74.30
	YU	9.83	30.70	51.07

Table D.5 – Y-Velocity Percent Error Results for the Z = 1.2 dm LDCF Cases

Case	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1 81x81x17	SBB	13.05	59.18	86.63
	BZ	7.27	32.59	54.84
	YU	8.40	26.61	46.95
LDCF2 101x101x21	SBB	12.38	52.42	86.88
	BZ	11.58	35.01	59.53
	YU	9.43	27.86	49.19
LDCF3 121x121x25	SBB	15.48	48.23	81.68
	BZ	13.99	40.69	69.66
	YU	10.89	30.35	52.26

Table D.6 – Z-Velocity Percent Error Results for the Z = 1.2 dm LDCF Cases

Case	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1 81x81x17	SBB	4.60	25.32	54.74
	BZ	3.50	16.85	28.82
	YU	2.47	12.11	23.68
LDCF2 101x101x21	SBB	4.94	29.96	48.12
	BZ	4.21	19.87	33.22
	YU	2.59	13.63	28.03
LDCF3 121x121x25	SBB	5.48	28.65	44.04
	BZ	4.52	24.88	37.34
	YU	2.63	15.51	32.17

Flow in an Annulus

<u>Absolute Error Visualization Plots for 3D Annulus Cases</u>

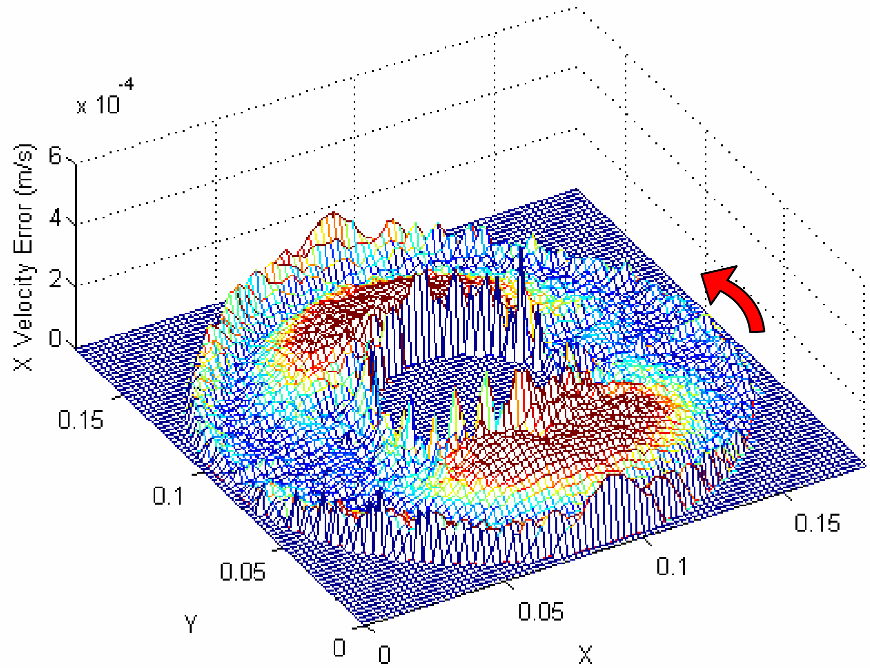


Figure D.21 – Typical X-Velocity Error for Z = 0.9dm Plane

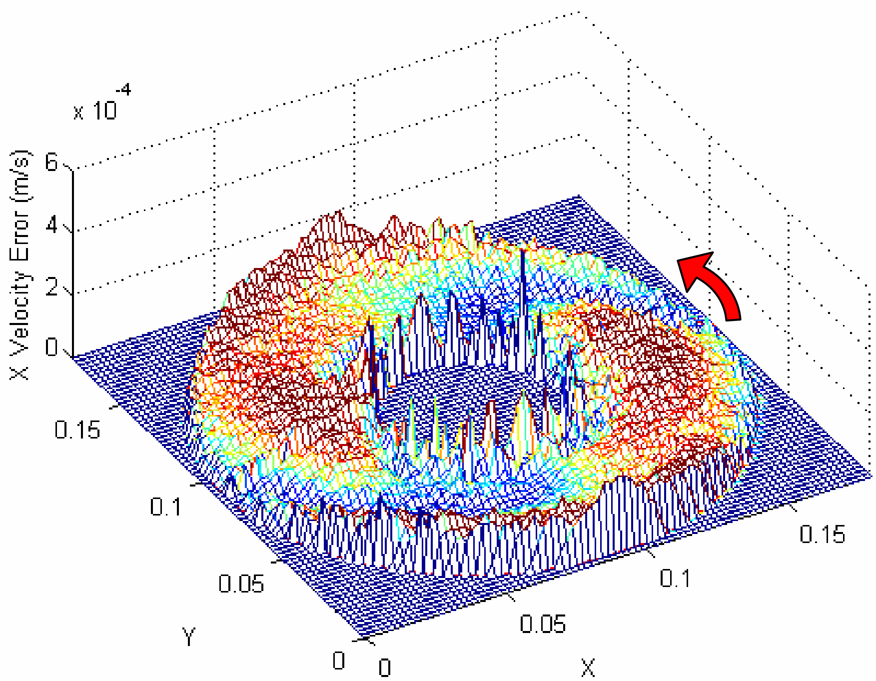


Figure D.22 – Typical X-Velocity Error for Z = 0.27dm Plane

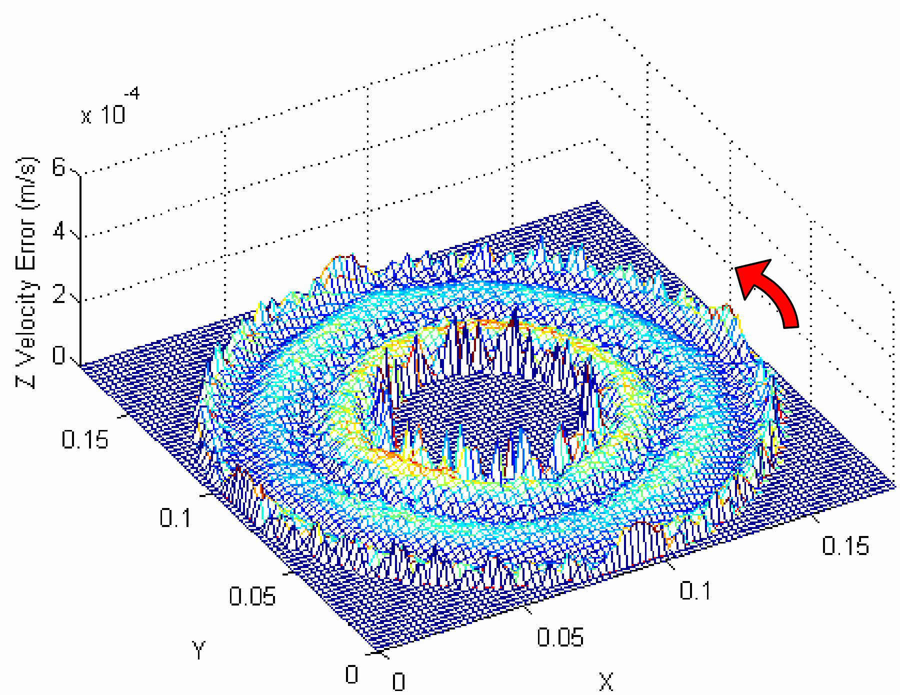


Figure D.23 – Typical Z-Velocity Error for Z = 0.27dm Plane

Velocity Line Plots at Z = 0.27dm, Y = 0.9dm for 3D Annulus Cases

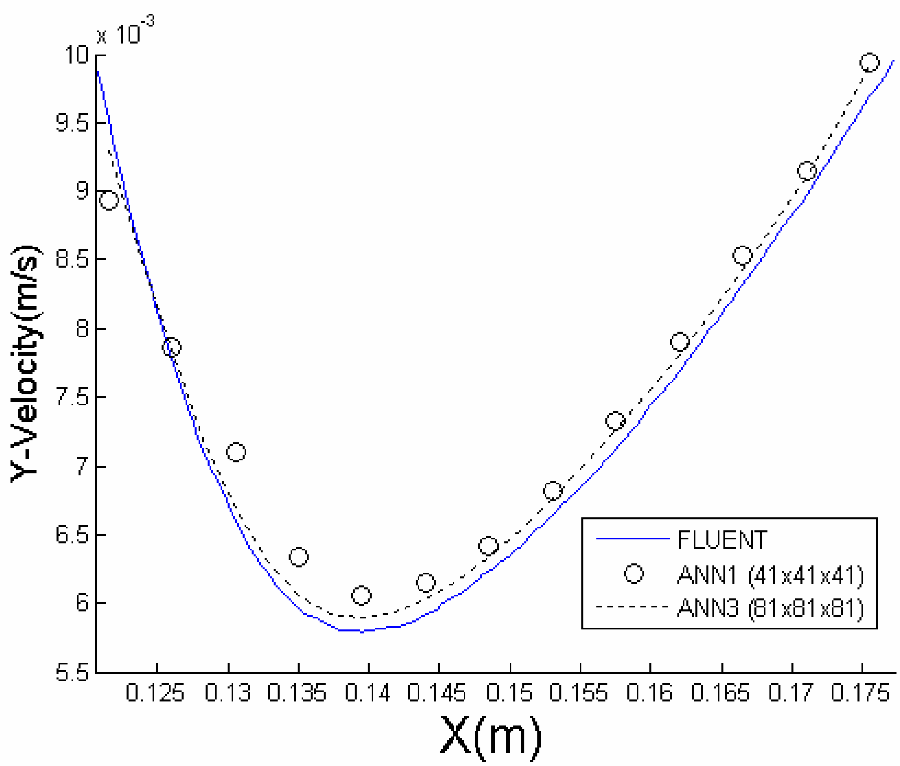


Figure D.24 – 3D ANN Y-Velocity Plot using the SBB Treatment

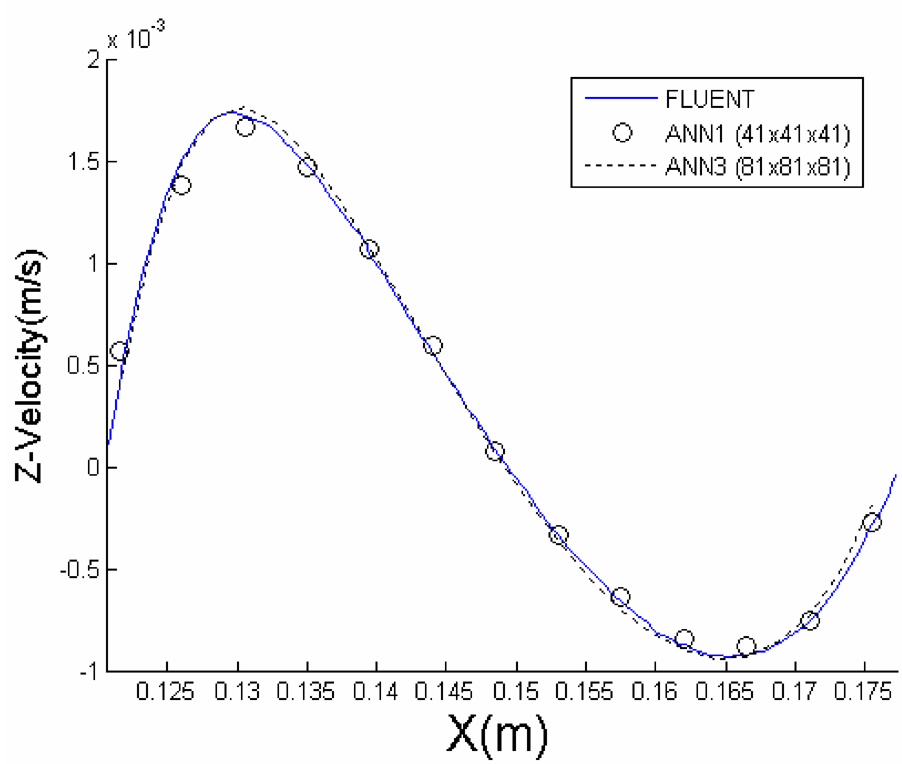


Figure D.25 – 3D ANN Z-Velocity Plot using the SBB Treatment

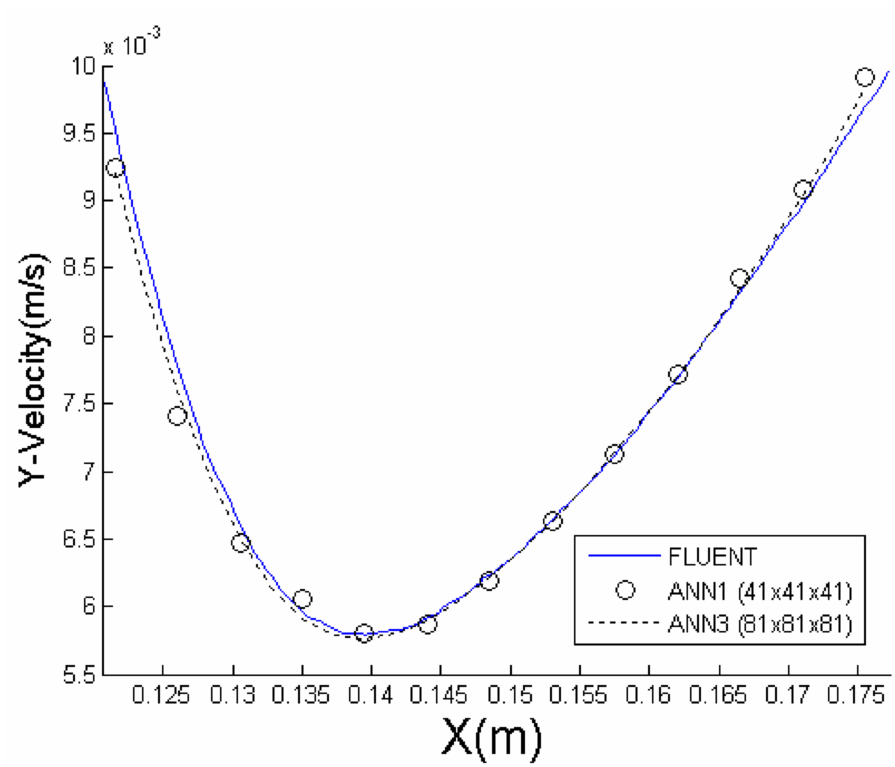


Figure D.26 – 3D ANN Y-Velocity Plot using the BZ Treatment

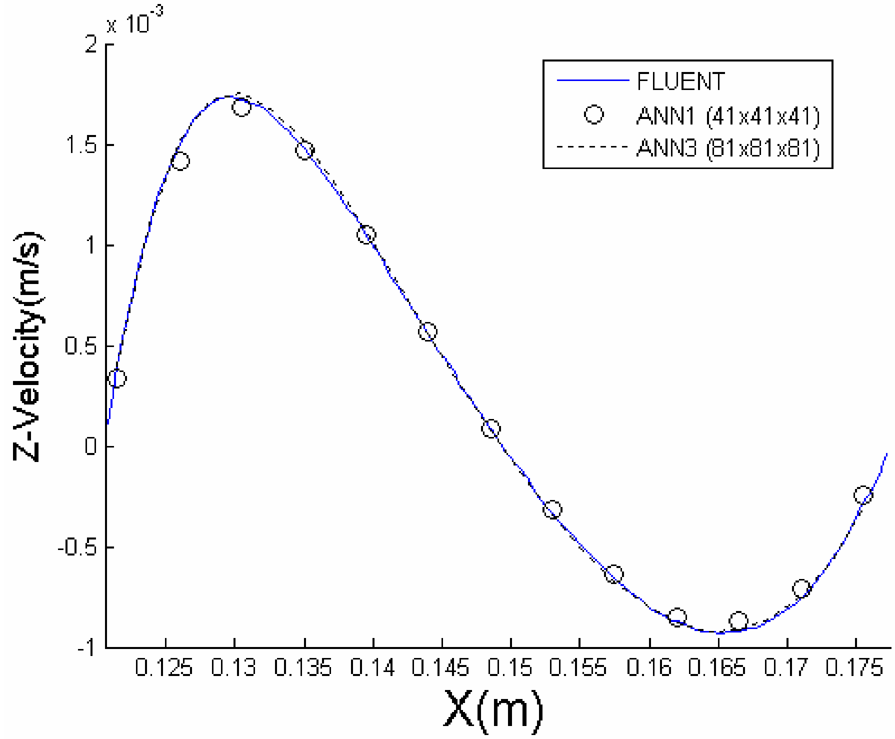


Figure D.27 – 3D ANN Z-Velocity Plot using the BZ Treatment

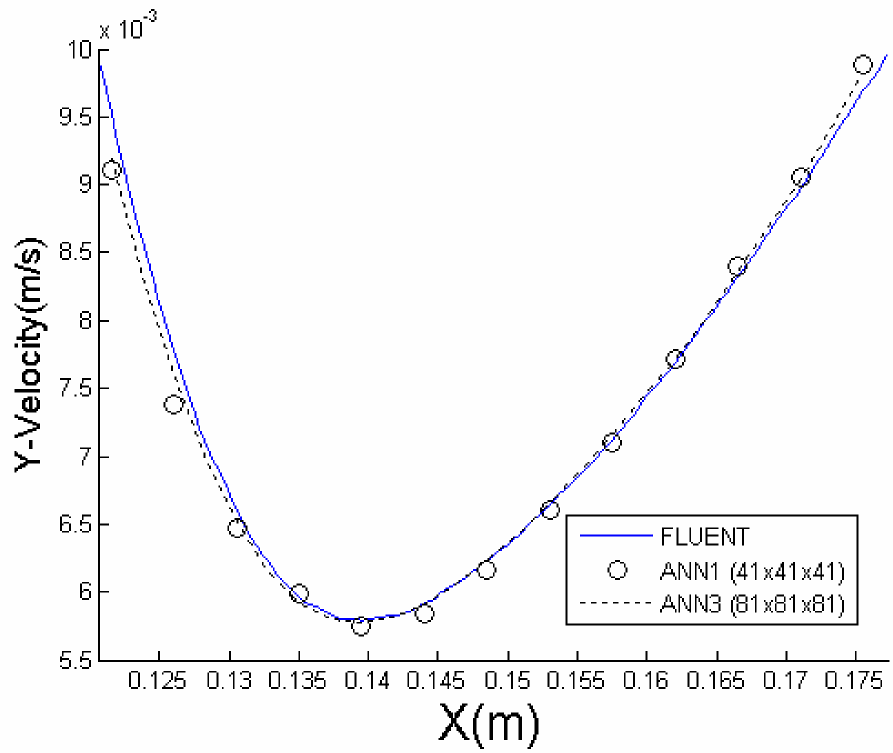


Figure D.28 – 3D ANN Y-Velocity Plot using the YU Treatment

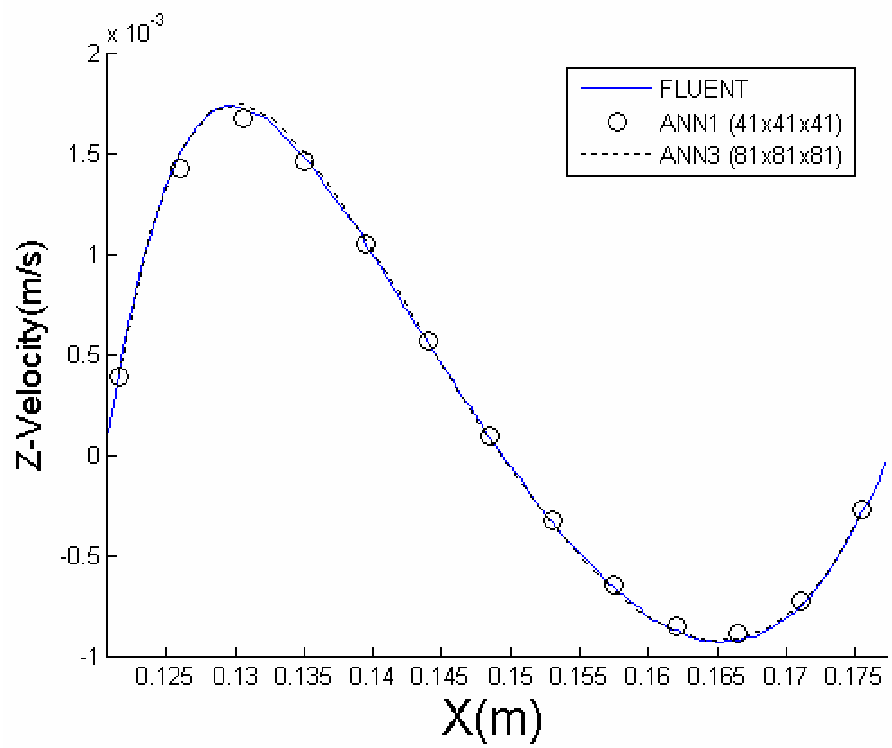


Figure D.29 – 3D ANN Z-Velocity Plot using the YU Treatment

RMS Error Plots for Annulus Cases

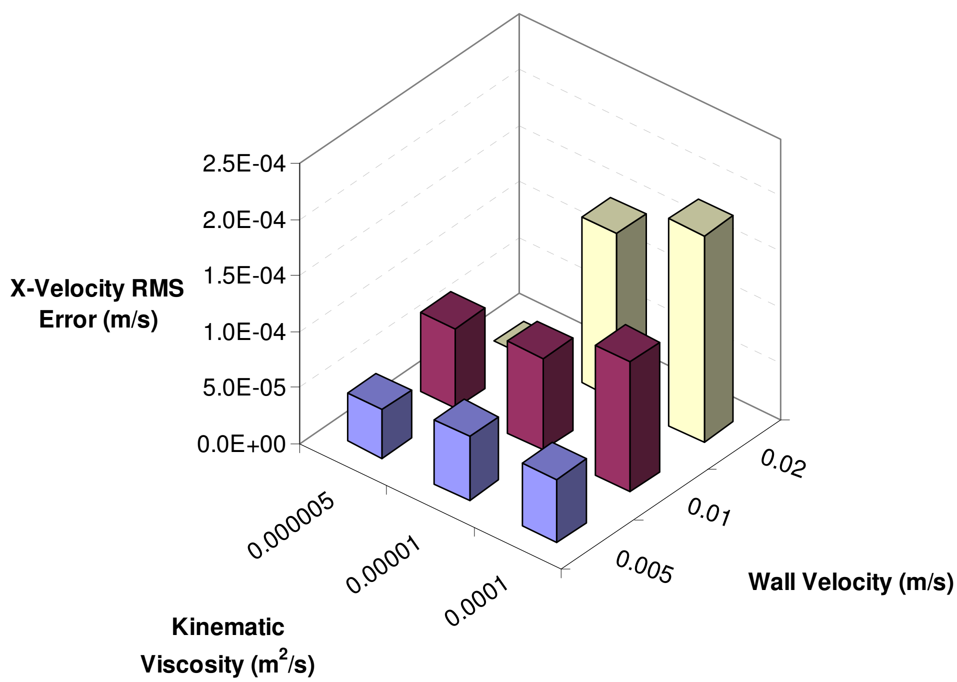


Figure D.30 – X-Velocity RMS Error for the 3D ANN3 grid using the SBB Treatment

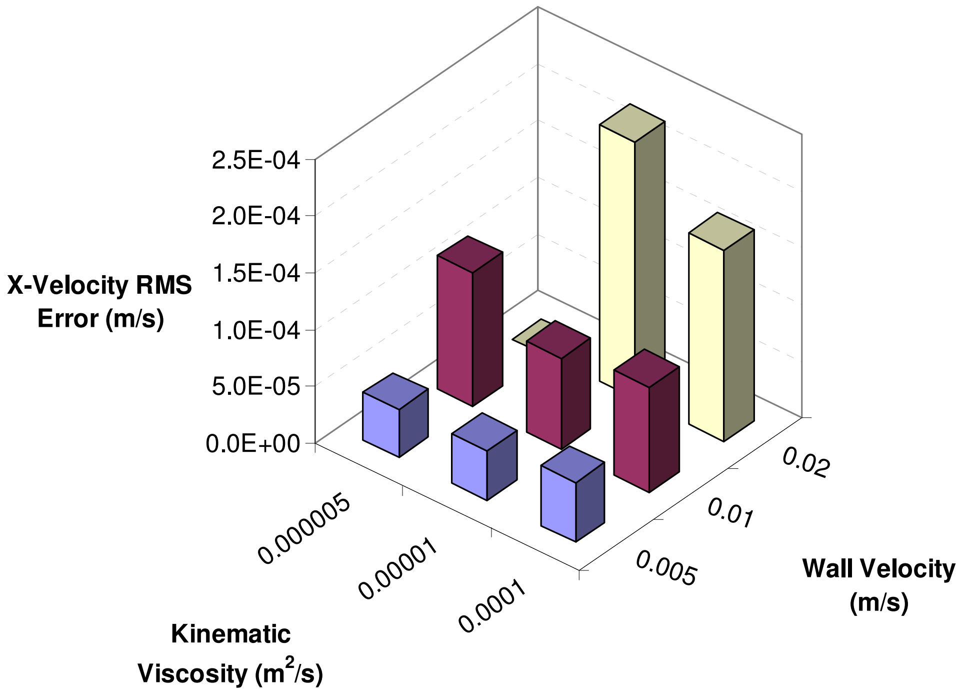


Figure D.31 – X-Velocity RMS Error for the ANN3 grid using the BZ Treatment

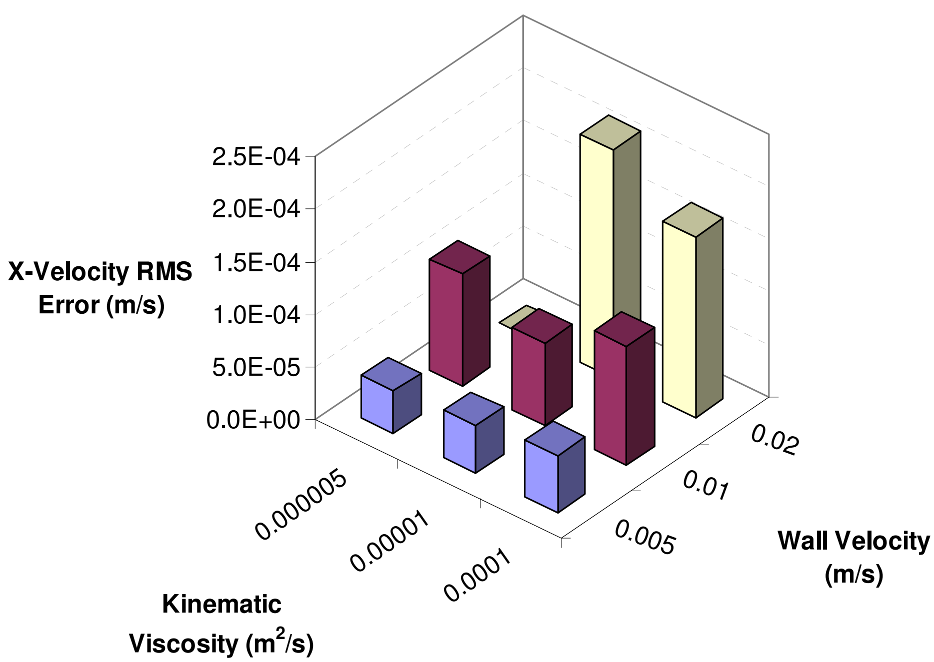


Figure D.32 – X-Velocity RMS Error for the ANN3 grid using the YU Treatment

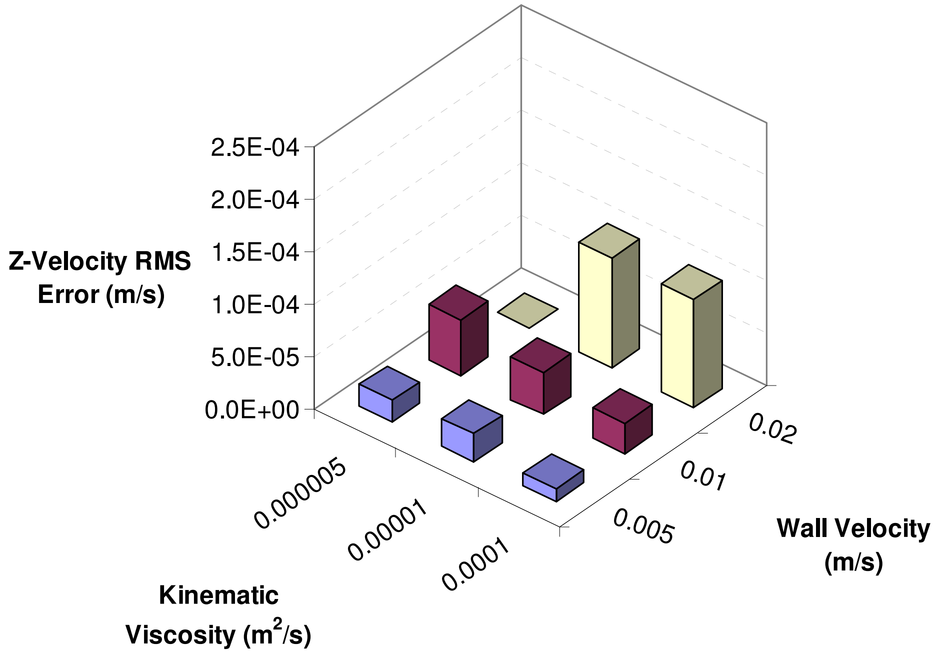


Figure D.33 – Z-Velocity RMS Error for the ANN3 grid using the SBB Treatment

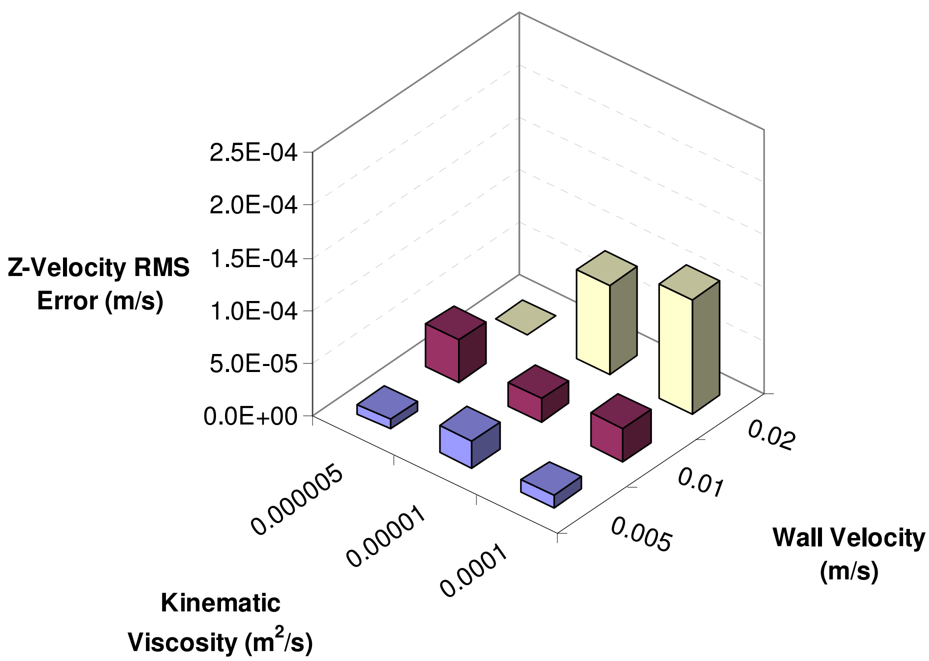


Figure D.34 – Z-Velocity RMS Error for the ANN3 grid using the BZ Treatment

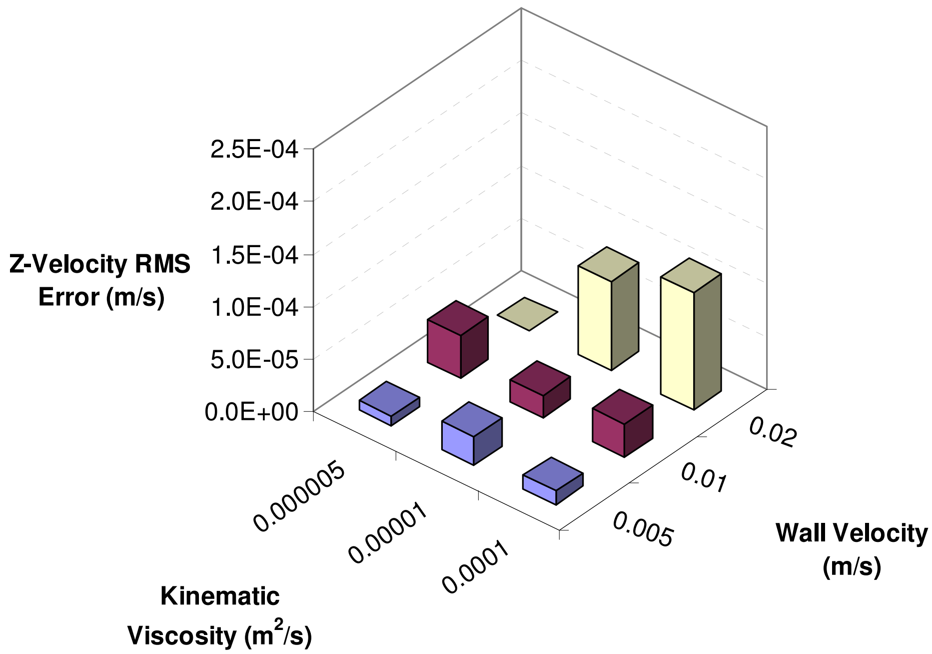


Figure D.35 – Z-Velocity RMS Error for the ANN3 grid using the YU Treatment

<u>Percent Error Results for 3D Annulus Cases at Z = 0.27dm</u>

Table D.7 – X-Velocity Percent Error Results for the 3D ANN-CAS5 Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-CAS5 41x41x41	SBB	13.92	77.46	88.92
	BZ	33.14	80.02	92.52
	YU	30.59	78.69	91.76
ANN2-CAS5 61x61x61	SBB	19.76	84.21	92.53
	BZ	40.20	85.18	93.07
	YU	40.79	85.94	93.20
ANN3-CAS5 81x81x81	SBB	34.40	86.79	93.38
	BZ	46.15	87.41	93.50
	YU	45.15	87.70	93.62

Table D.8 – Z-Velocity Percent Error Results for the 3D ANN-CAS5 Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-CAS5 41x41x41	SBB	13.45	52.18	67.52
	BZ	10.51	53.50	80.49
	YU	15.34	63.54	84.47
ANN2-CAS5 61x61x61	SBB	12.88	65.25	80.41
	BZ	34.04	81.33	90.63
	YU	31.29	78.89	89.02
ANN3-CAS5 81x81x81	SBB	11.36	65.08	81.54
	BZ	32.29	84.90	93.29
	YU	28.26	84.61	92.40

Absolute Error Visualization Plots for 3D Annulus Construct Cases

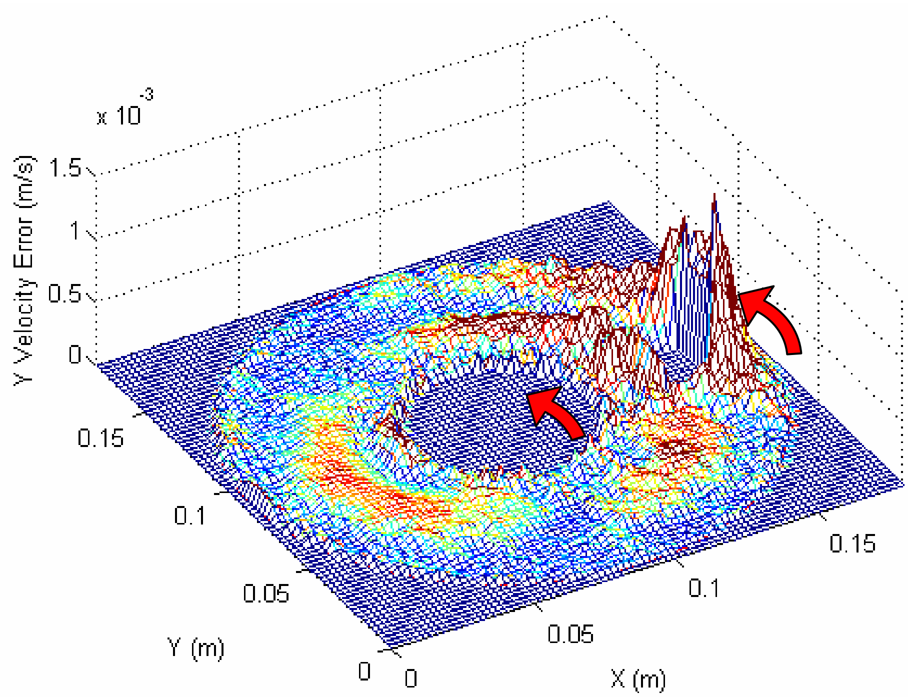


Figure D.36 – Typical Y-Velocity Error for 3D Annulus Construct Case

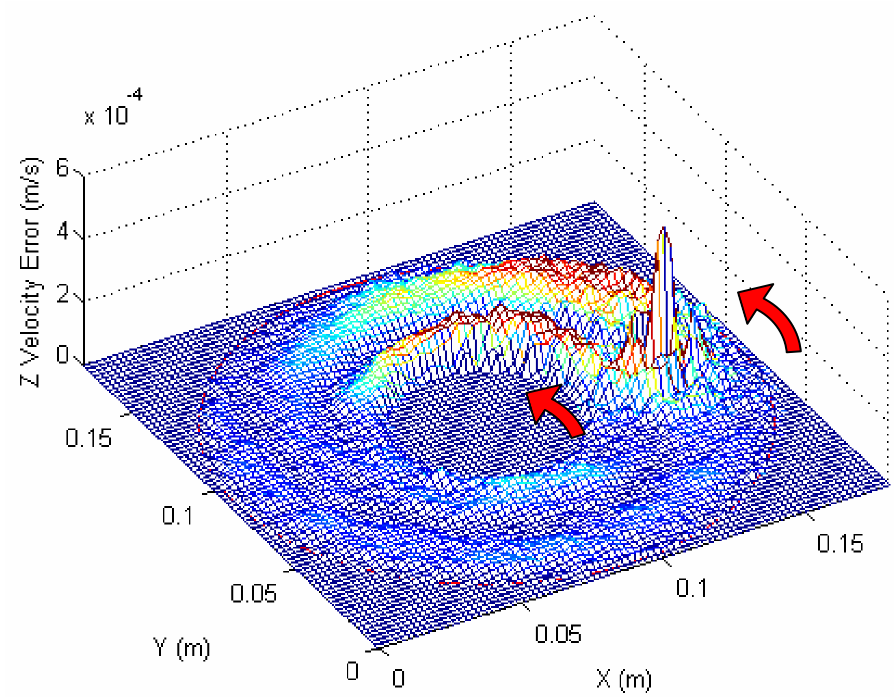


Figure D.37 – Typical Z-Velocity Error for 3D Annulus Construct Case

RMS Error Plots at Z = 0.81dm for 3D Annulus Construct Cases

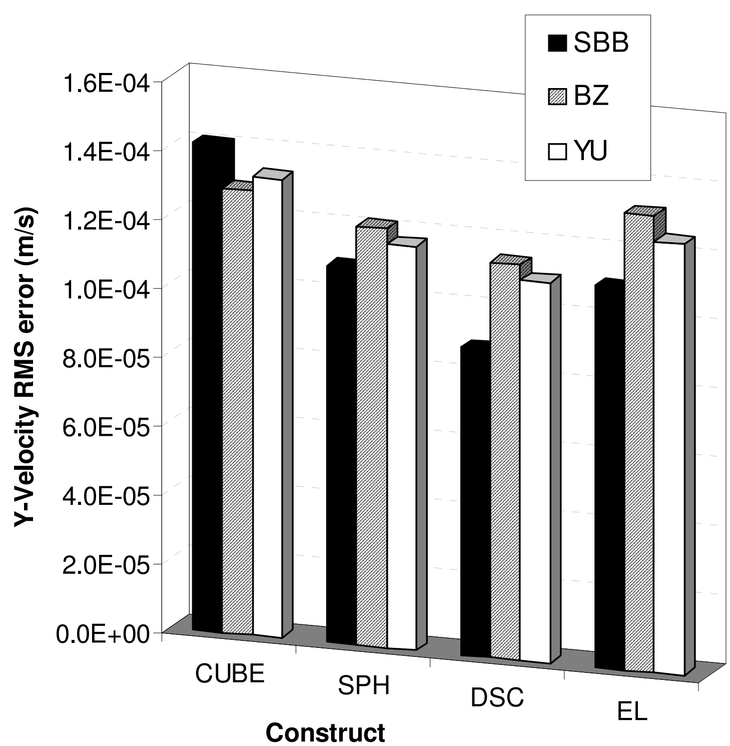


Figure D.38 – Y-Velocity RMS Error versus Construct for ANN3-CAS5 Cases

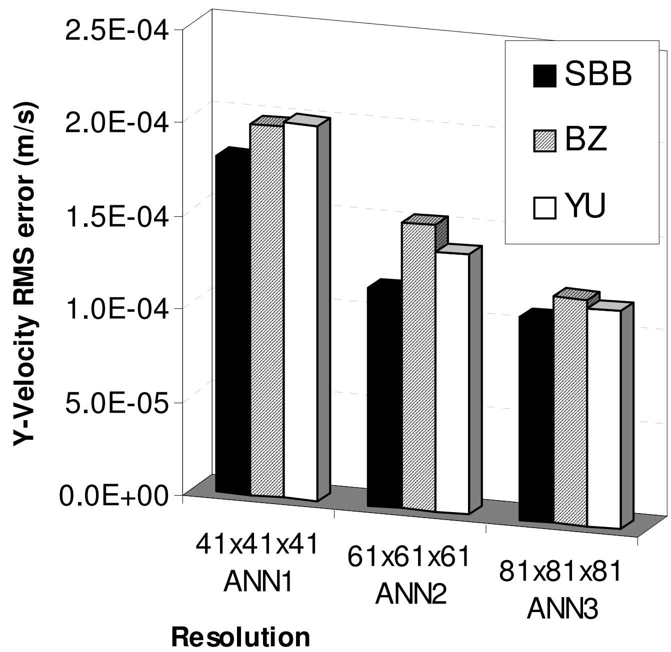


Figure D.39 – Y-Velocity RMS Error versus Resolution for ANN-SPH-CAS5 Cases

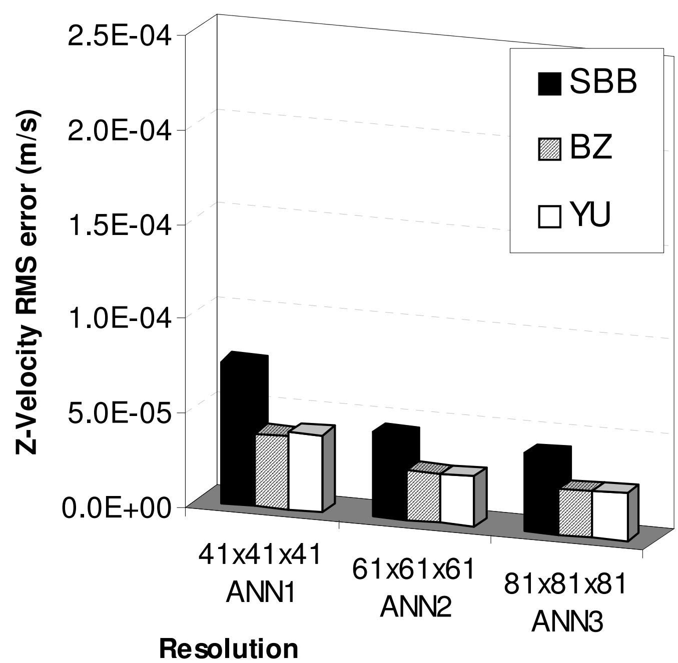


Figure D.40 – Z-Velocity RMS Error versus Resolution for ANN-SPH-CAS5 Cases

Percent Error Results at Z = 0.81dm for 3D Annulus Construct Cases

Table D.9 – Y-Velocity Percent Error Results for the 3D ANN3-CAS5 Construct Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN3-CAS5 CUBE	SBB	46.21	87.26	94.57
	BZ	26.58	86.09	94.25
	YU	29.36	88.22	94.84
ANN3-CAS5 SPH	SBB	42.57	86.36	93.48
	BZ	26.93	86.02	95.65
	YU	28.78	86.60	95.77
ANN3-CAS5 DSC	SBB	44.16	87.62	95.45
	BZ	29.11	88.32	95.69
	YU	30.60	89.72	96.20
ANN3-CAS5 EL	SBB	42.47	85.95	94.20
	BZ	25.74	84.06	95.52
	YU	27.40	86.00	95.88

Table D.10 – Y-Velocity Percent Error Results for the 3D ANN-SPH-CAS5 Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-CAS5 SPH	SBB	34.13	80.70	89.54
	BZ	21.39	69.20	92.49
	YU	21.20	69.01	92.68
ANN2-CAS5 SPH	SBB	43.01	85.09	93.14
	BZ	23.14	81.10	94.92
	YU	26.19	84.03	95.42
ANN3-CAS5 SPH	SBB	42.57	86.36	93.48
	BZ	26.93	86.02	95.65
	YU	28.78	86.60	95.77

Table D.11 – Z-Velocity Percent Error Results for the 3D ANN-SPH-CAS5 Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-CAS5 SPH	SBB	5.44	23.09	38.93
	BZ	5.73	29.58	53.91
	YU	5.63	29.01	53.24
ANN2-CAS5 SPH	SBB	3.74	18.83	40.50
	BZ	5.01	26.69	51.59
	YU	5.10	25.54	50.45
ANN3-CAS5 SPH	SBB	2.71	16.43	37.48
	BZ	3.85	23.06	45.10
	YU	4.81	22.70	45.42

<u>Absolute Error Visualization Plots for the Refined LDCF Cases at Z = 1.2dm</u>

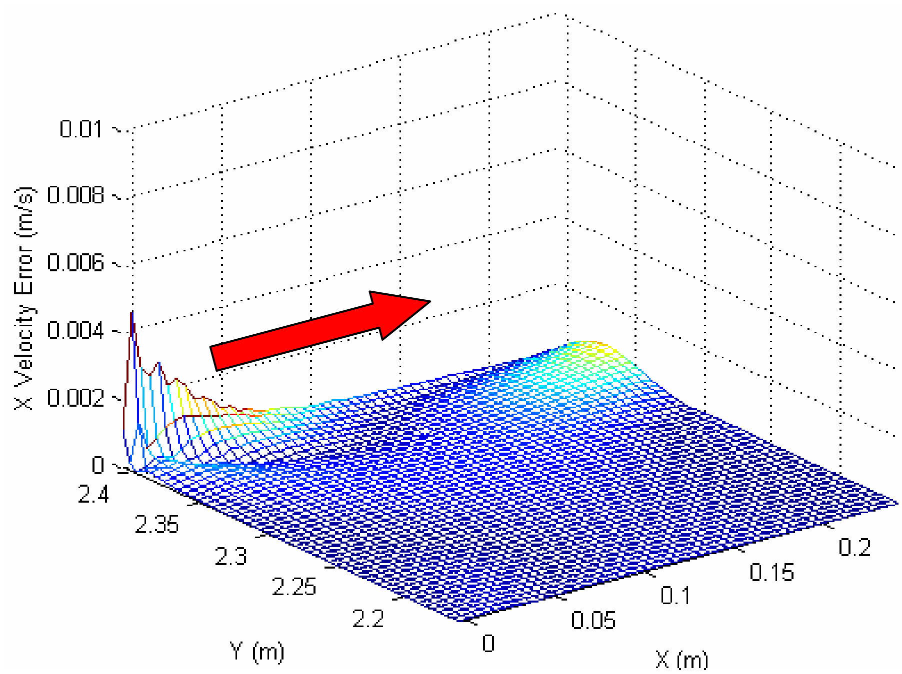


Figure D.41 – Absolute X-Velocity Error for the LDCF3-RF4-UL Case

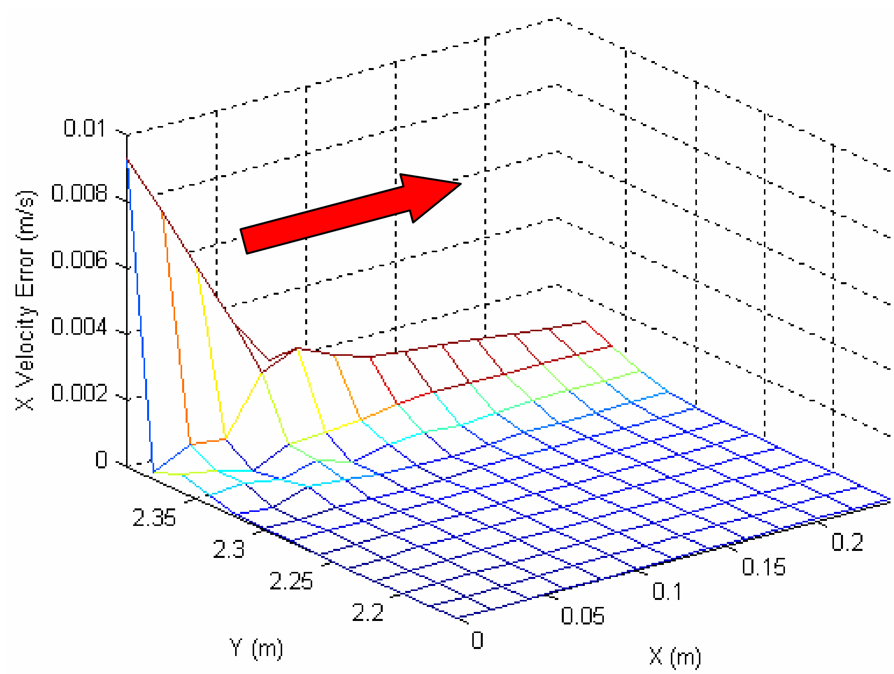


Figure D.42 – Absolute X-Velocity Error for the LDCF3 Case

RMS Error Plots for the Refined 3D LDCF Cases

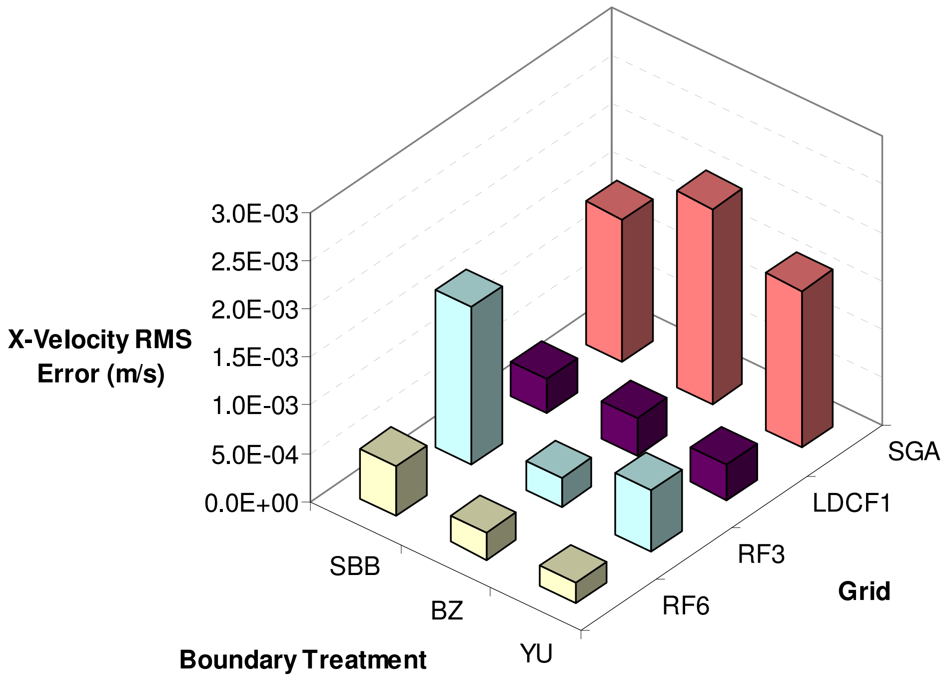


Figure D.43 – X-Velocity RMS Error for the LDCF1-UL Grid Refinement

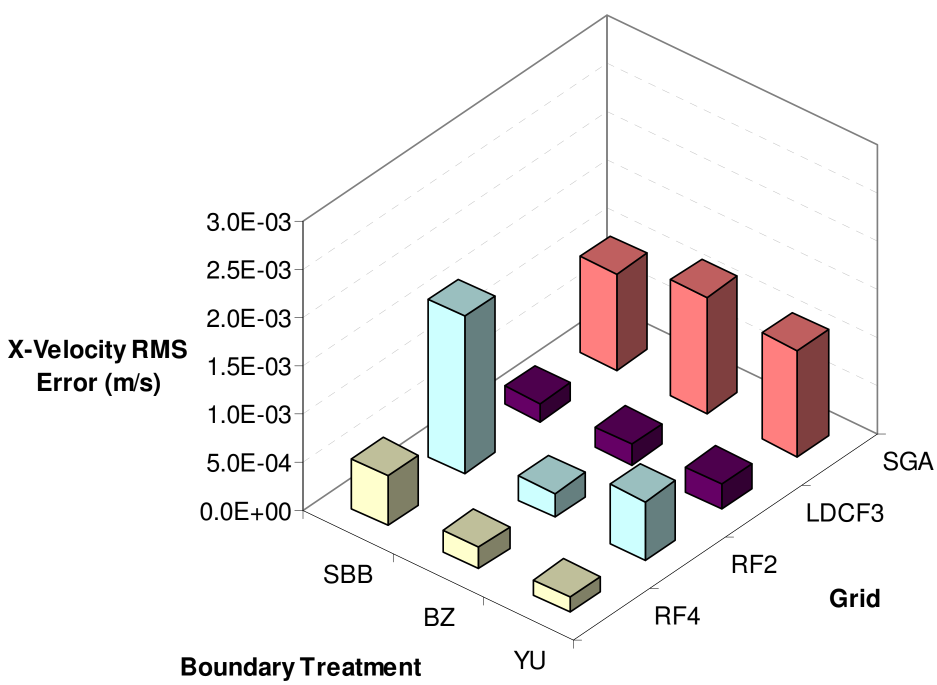


Figure D.44 – X-Velocity RMS Error for the LDCF3-UL Grid Refinement

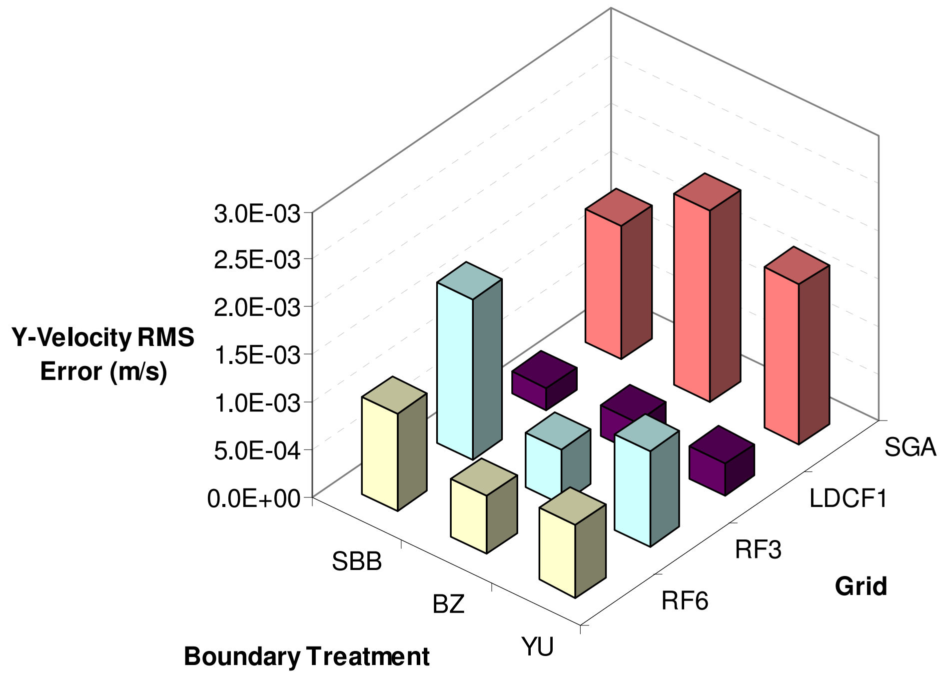


Figure D.45 – Y-Velocity RMS Error for the LDCF1-UR Grid Refinement

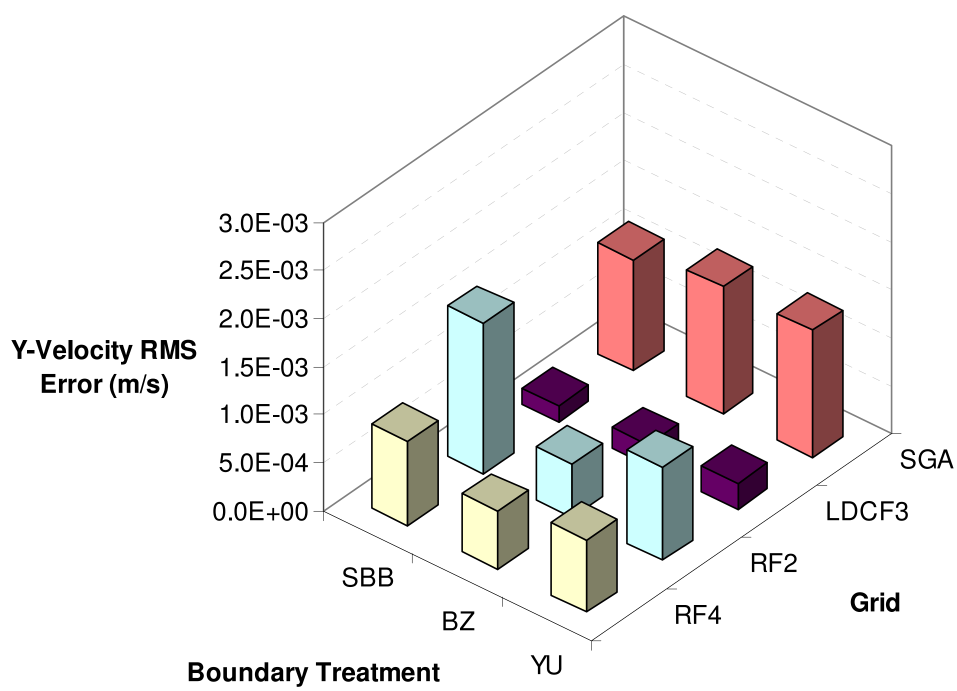


Figure D.46 – Y-Velocity RMS Error for the LDCF3-UR Grid Refinement

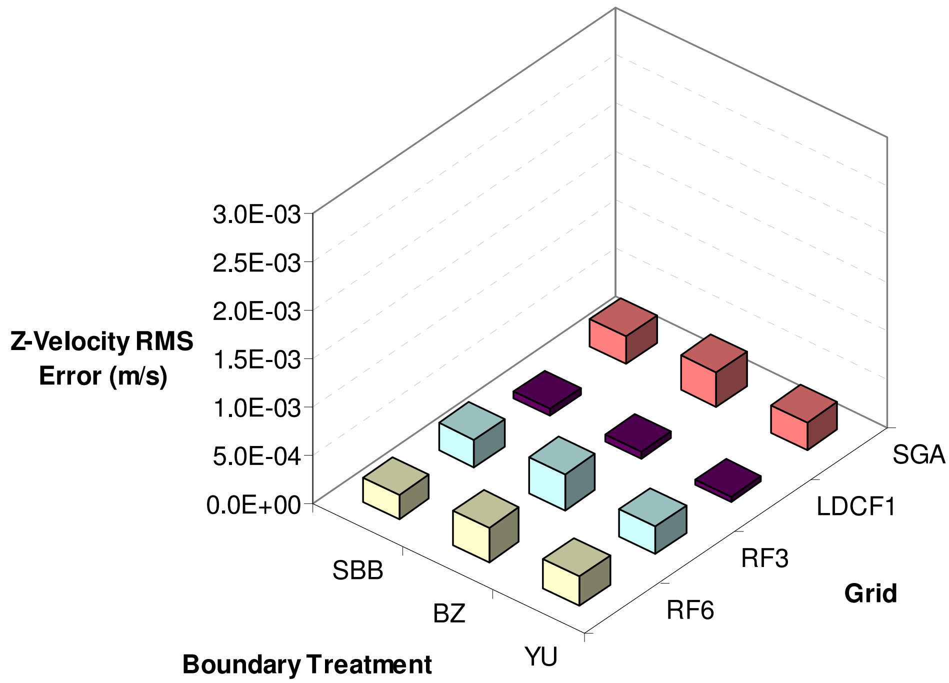


Figure D.47 – Z-Velocity RMS Error for the LDCF1-UR Grid Refinement

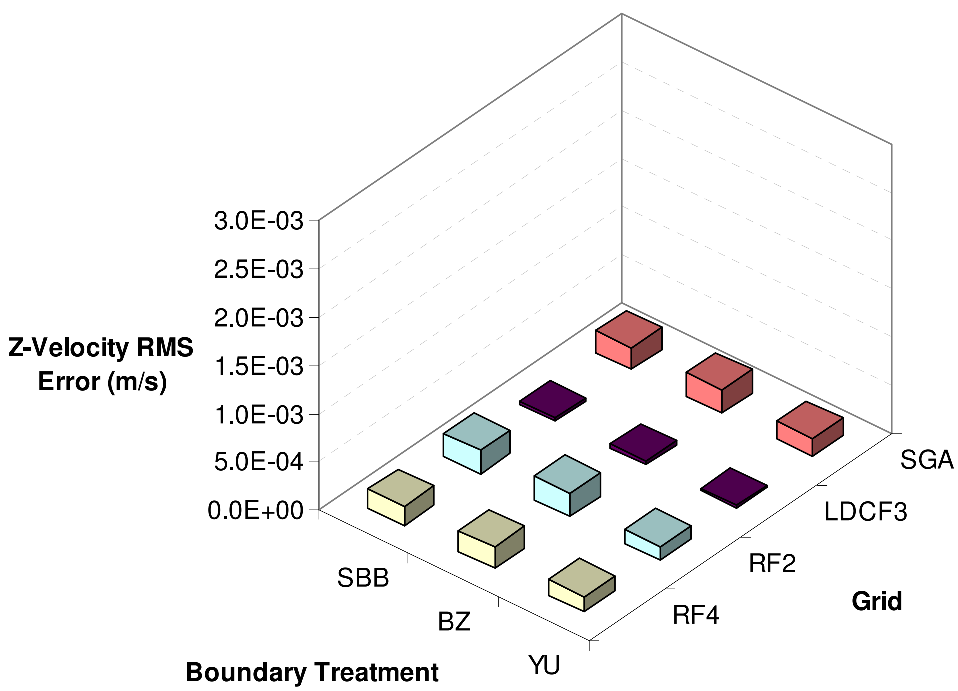


Figure D.48 – Z-Velocity RMS Error for the LDCF3-UR Grid Refinement

RMS Error Results for the Refined 3D LDCF Cases

Table D.12 – Percent Decrease in X-Velocity RMS Error for 3D LDCF-UL Cases

Grid	Boundary Treatment	RMS error X-Velocity Sub-Grid (m/s)	RMS error X-Velocity SGA (m/s)	% Decrease in RMS error
LDCF1-UL RF6	SBB	0.000527	0.001482	64.42%
	BZ	0.000281	0.002039	86.20%
	YU	0.000211	0.001636	87.07%
LDCF3-UL RF4	SBB	0.000517	0.000996	48.09%
	BZ	0.000227	0.001207	81.16%
	YU	0.000161	0.001110	85.51%

Table D.13 – Percent Decrease in Y-Velocity RMS Error for 3D LDCF-UR Cases

Grid	Boundary Treatment	RMS error Y-Velocity Sub-Grid (m/s)	RMS error Y-Velocity SGA (m/s)	% Decrease in RMS error
LDCF1-UR RF6	SBB	0.001028	0.001388	25.93%
	BZ	0.000611	0.002006	69.54%
	YU	0.000767	0.001675	54.19%
LDCF3-UR RF4	SBB	0.000878	0.001152	23.76%
	BZ	0.000597	0.001324	54.88%
	YU	0.000752	0.001321	43.05%

Table D.14 – Percent Decrease in Z-Velocity RMS Error for 3D LDCF-UR Cases

Grid	Boundary Treatment	RMS error Z-Velocity Sub-Grid (m/s)	RMS error Z-Velocity SGA (m/s)	% Decrease in RMS error
LDCF1-UR RF6	SBB	0.000240	0.000286	15.97%
	BZ	0.000351	0.000352	0.45%
	YU	0.000303	0.000295	-2.92%
LDCF3-UR RF4	SBB	0.000191	0.000222	13.96%
	BZ	0.000220	0.000236	6.50%
	YU	0.000160	0.000185	13.67%

Percent Error Results for the Refined 3D LDCF Cases

Table D.15 – X-Velocity Percent Error Results for the 3D LDCF-UL Sub-Grid Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1-UL RF6	SBB	5.68	35.33	70.95
	BZ	10.64	41.86	81.59
	YU	10.76	37.02	81.63
LDCF3-UL RF4	SBB	15.05	60.18	82.67
	BZ	19.89	73.35	87.48
	YU	11.89	72.15	87.32

Table D.16 – X-Velocity Percent Error Results for the 3D LDCF-UL SGA Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1-UL SGA	SBB	4.94	24.69	48.15
	BZ	8.64	33.33	49.38
	YU	0.00	24.69	51.85
LDCF3-UL SGA	SBB	6.51	59.76	75.74
	BZ	1.18	55.62	72.78
	YU	1.18	8.28	66.86

Table D.17 – Y-Velocity Percent Error Results for the 3D LDCF-UR Sub-Grid Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1-UR RF6	SBB	0.00	20.18	40.23
	BZ	12.29	43.48	57.53
	YU	6.65	31.99	53.16
LDCF3-UR RF4	SBB	10.97	31.31	49.44
	BZ	10.41	43.52	58.09
	YU	9.73	32.47	54.12

Table D.18 – Y-Velocity Percent Error Results for the 3D LDCF-UR SGA Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1-UR SGA	SBB	2.47	25.93	56.79
	BZ	9.88	39.51	49.38
	YU	6.17	29.63	46.91
LDCF3-UR SGA	SBB	8.28	35.50	57.40
	BZ	13.61	34.32	49.70
	YU	1.18	25.44	45.56

Table D.19 – Z-Velocity Percent Error Results for the 3D LDCF-UR Sub-Grid Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1-UR RF6	SBB	3.78	17.32	33.68
	BZ	1.37	10.61	22.47
	YU	0.88	4.86	21.74
LDCF3-UR RF4	SBB	5.47	20.78	36.86
	BZ	4.26	21.38	38.38
	YU	5.35	26.97	48.59

Table D.20 – Z-Velocity Percent Error Results for the 3D LDCF-UR SGA Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
LDCF1-UR SGA	SBB	3.80	17.72	32.91
	BZ	2.53	17.72	35.44
	YU	1.27	11.39	32.91
LDCF3-UR SGA	SBB	8.33	33.33	51.19
	BZ	4.17	29.76	44.64
	YU	3.57	26.79	54.17

RMS Error Results for the 3D LDCF-LM Cases

Table D.21 – RMS X-Velocity Error Results for 3D LDCF-LM Cases

Grid	Boundary Treatment	RMS error X-Velocity Sub-Grid (m/s)	RMS error X-Velocity Main Grid (m/s)
LDCF1-LM RF6	SBB	0.000174	0.000162
	BZ	0.000353	0.000340
	YU	0.000450	0.000438
LDCF3-LM RF4	SBB	0.000219	0.000213
	BZ	0.000273	0.000267
	YU	0.000382	0.000376

Table D.22 – RMS Z-Velocity Error Results for 3D LDCF-LM Cases

Grid	Boundary Treatment	RMS error Z-Velocity Sub-Grid (m/s)	RMS error Z-Velocity Main Grid (m/s)
LDCF1-LM RF6	SBB	0.000017	0.000006
	BZ	0.000019	0.000014
	YU	0.000021	0.000017
LDCF3-LM RF4	SBB	0.000017	0.000008
	BZ	0.000018	0.000010
	YU	0.000019	0.000014

Absolute Error Visualization Plots for the 3D Refined Annulus Cases at Z = 0.9dm

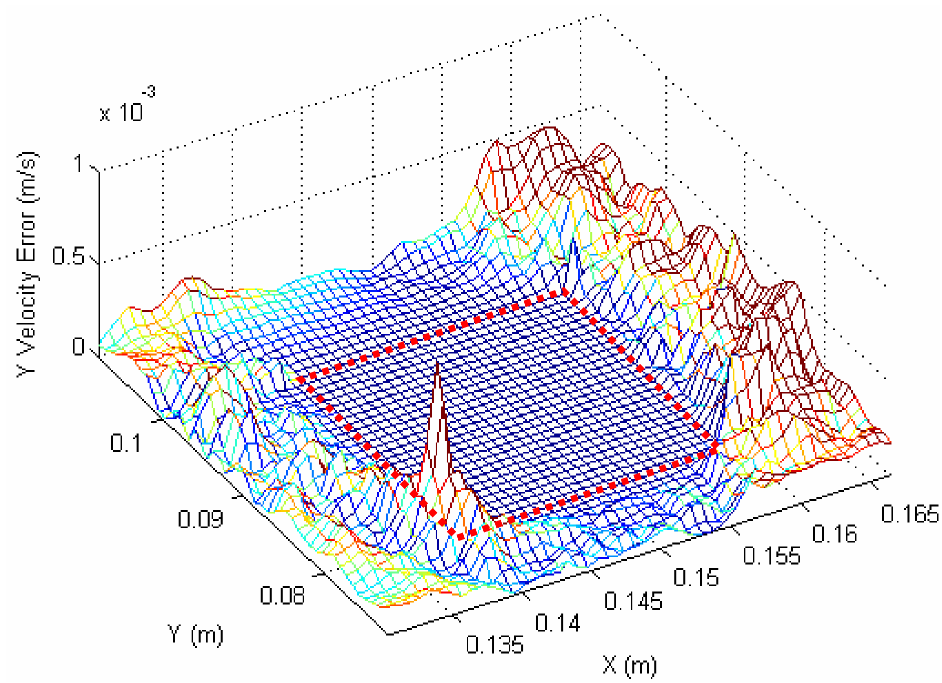


Figure D.49 – Absolute Y-Velocity Error for the ANN3-RF3-CUBE Case

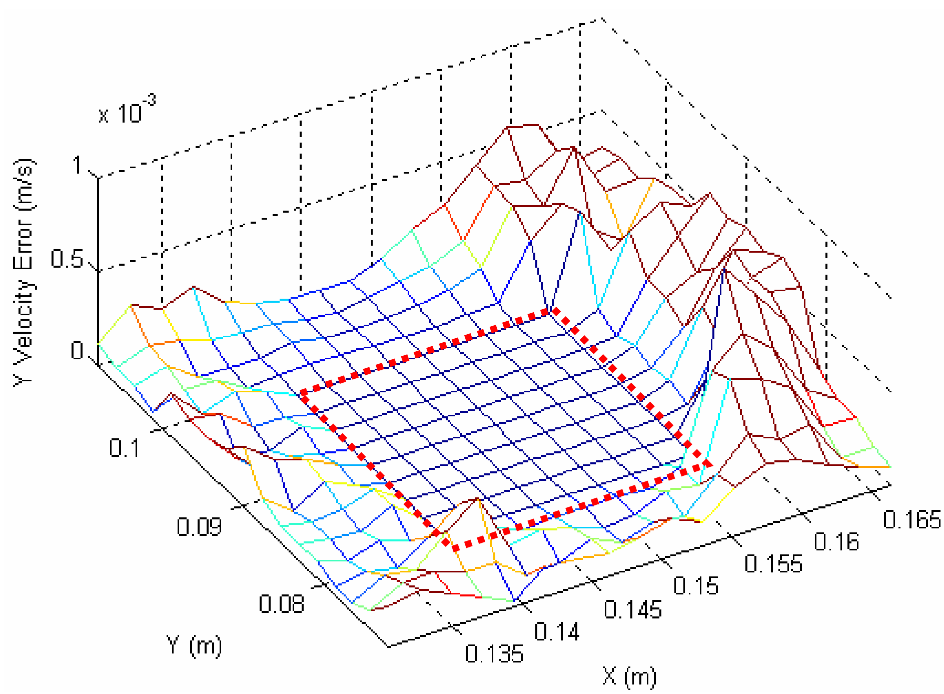


Figure D.50 – Absolute Y-Velocity Error for the ANN3-CUBE Case

RMS Error Plots for the 3D Refined Annulus Cases

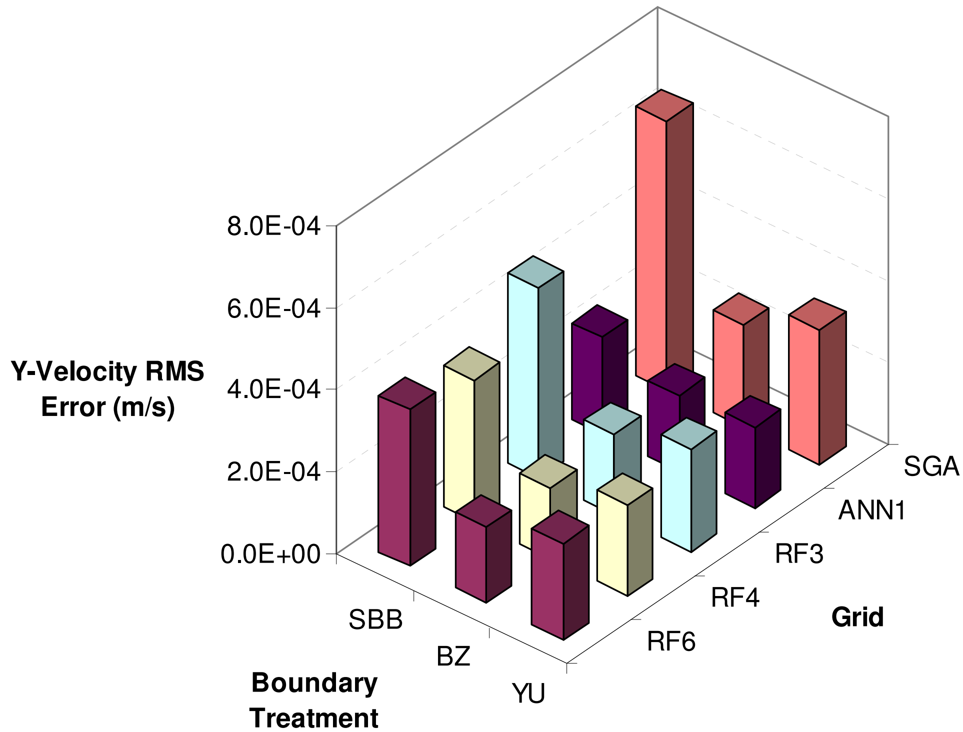


Figure D.51 – Y-Velocity RMS Error for the ANN1-CUBE Grid Refinement

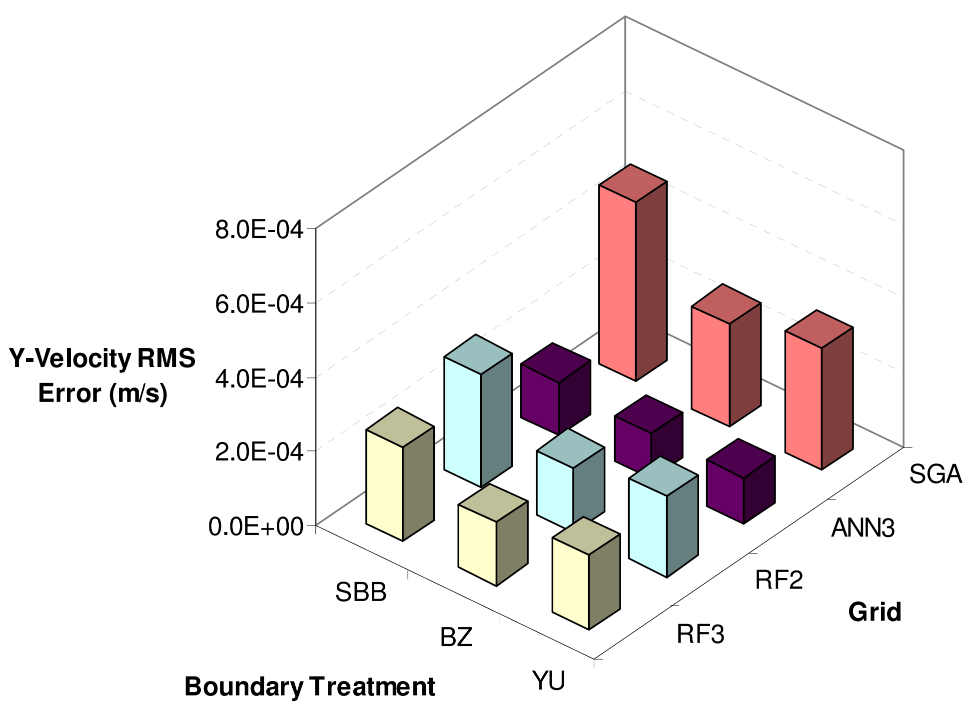


Figure D.52 – Y-Velocity RMS Error for the ANN3-CUBE Grid Refinement

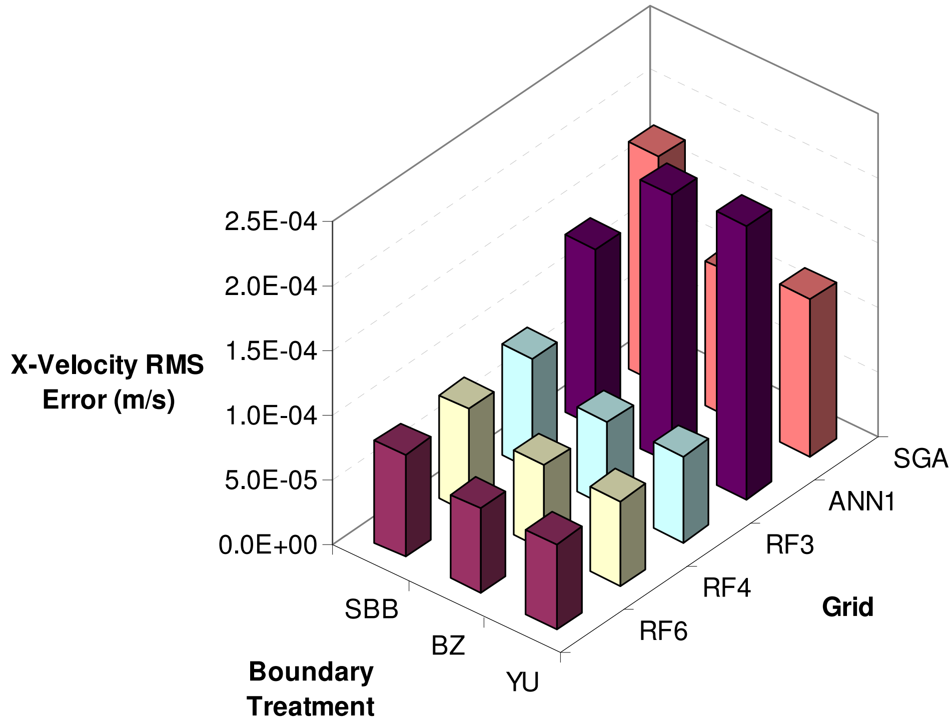


Figure D.53 – X-Velocity RMS Error for the ANN1-SPH Grid Refinement

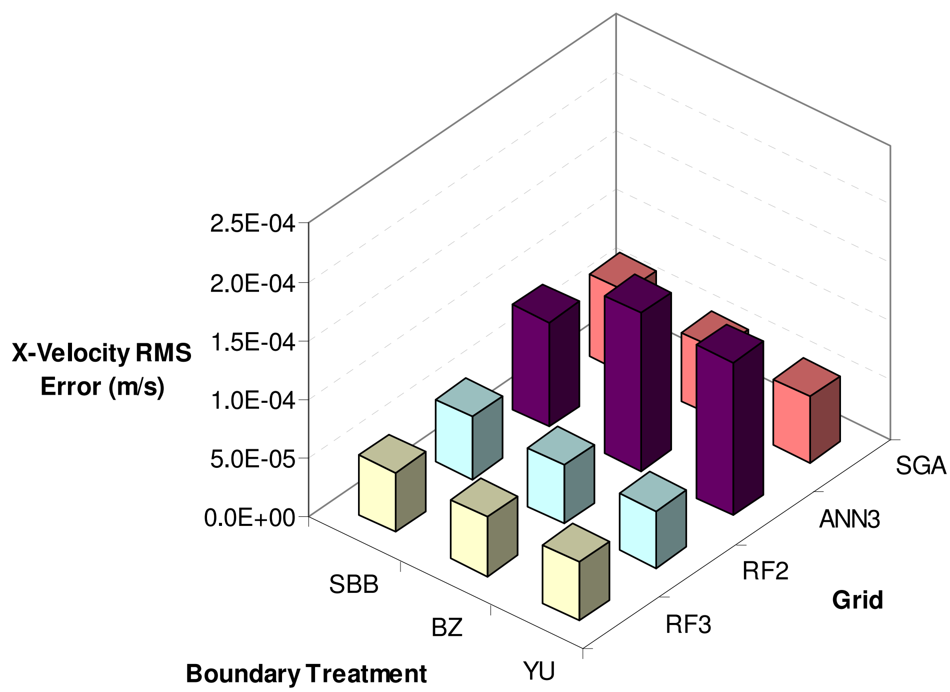


Figure D.54 – X-Velocity RMS Error for the ANN3-SPH Grid Refinement

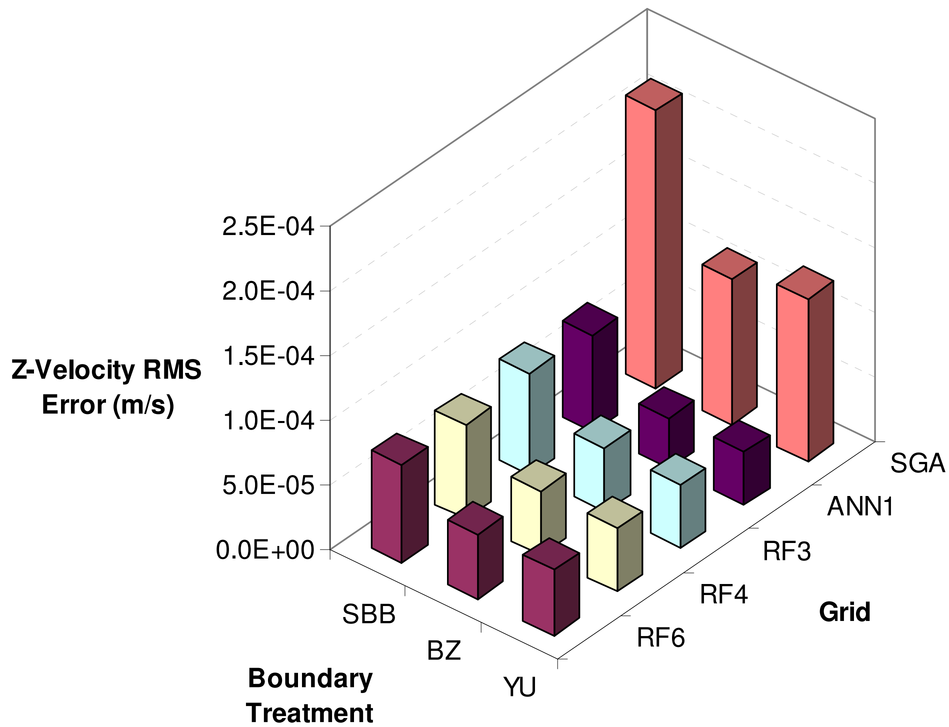


Figure D.55 – Z-Velocity RMS Error for the ANN1-SPH Grid Refinement

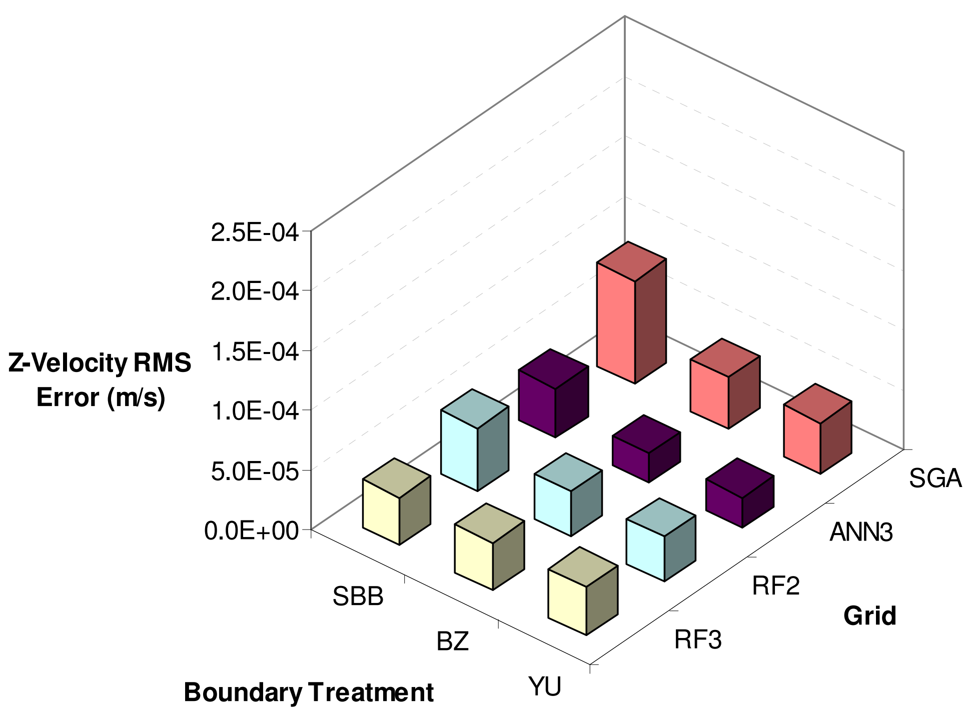


Figure D.56 – Z-Velocity RMS Error for the ANN3-SPH Grid Refinement

Absolute Error Results for the 3D Refined Annulus Cases

Table D.23 – Percent Decrease in Y-Velocity RMS Error for 3D ANN-CUBE Cases

Grid	Boundary Treatment	RMS error Y-Velocity Sub-Grid (m/s)	RMS error Y-Velocity SGA (m/s)	% Decrease in RMS error
ANN1-CUBE RF6	SBB	0.000382	0.000657	41.88%
	BZ	0.000185	0.000249	25.78%
	YU	0.000236	0.000328	27.93%
ANN2-CUBE RF4	SBB	0.000193	0.000492	60.74%
	BZ	0.000110	0.000150	26.49%
	YU	0.000111	0.000185	39.94%
ANN3-CUBE RF3	SBB	0.000253	0.000483	47.56%
	BZ	0.000170	0.000276	38.40%
	YU	0.000201	0.000328	38.66%

Table D.24 – Percent Decrease in X-Velocity RMS Error for 3D ANN-SPH Cases

Grid	Boundary Treatment	RMS error X-Velocity Sub-Grid (m/s)	RMS error X-Velocity SGA (m/s)	% Decrease in RMS error
ANN1-SPH RF6	SBB	0.000079	0.000176	55.12%
	BZ	0.000065	0.000112	41.89%
	YU	0.000066	0.000121	45.59%
ANN2-SPH RF4	SBB	0.000050	0.000093	46.53%
	BZ	0.000062	0.000081	23.17%
	YU	0.000056	0.000072	22.30%
ANN3-SPH RF3	SBB	0.000051	0.000071	28.30%
	BZ	0.000051	0.000059	13.67%
	YU	0.000049	0.000056	12.02%

Table D.25 – Percent Decrease in Z-Velocity RMS Error for 3D ANN-SPH Cases

Grid	Boundary Treatment	RMS error Z-Velocity Sub-Grid (m/s)	RMS error Z-Velocity SGA (m/s)	% Decrease in RMS error
ANN1-SPH RF6	SBB	0.000075	0.000215	64.94%
	BZ	0.000050	0.000112	55.54%
	YU	0.000051	0.000126	59.76%
ANN2-SPH RF4	SBB	0.000055	0.000101	45.76%
	BZ	0.000042	0.000063	32.11%
	YU	0.000041	0.000059	30.66%
ANN3-SPH RF3	SBB	0.000040	0.000086	54.05%
	BZ	0.000040	0.000043	8.27%
	YU	0.000041	0.000043	5.42%

Percent Error Results for the 3D Refined Annulus Cases

Table D.26 – Y-Velocity Percent Error Results for the 3D ANN-CUBE Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-CUBE RF6	SBB	4.87	25.02	41.51
	BZ	11.81	47.05	73.16
	YU	13.94	41.27	68.66
ANN2-CUBE RF4	SBB	11.08	46.62	65.98
	BZ	19.23	66.53	77.79
	YU	19.90	66.40	77.42
ANN3-CUBE RF3	SBB	9.43	33.84	63.66
	BZ	8.70	55.93	76.08
	YU	8.09	45.28	70.91

Table D.27 – Y-Velocity Percent Error Results for the 3D ANN-CUBE SGA Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-CUBE SGA	SBB	5.36	25.00	39.29
	BZ	10.71	48.21	64.29
	YU	3.57	42.86	53.57
ANN2-CUBE SGA	SBB	5.00	21.67	45.00
	BZ	20.83	63.33	74.17
	YU	10.00	54.17	73.33
ANN3-CUBE SGA	SBB	2.91	23.30	36.89
	BZ	7.28	40.29	63.59
	YU	6.80	36.41	54.37

Table D.28 – X-Velocity Percent Error Results for the 3D ANN-SPH Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-SPH RF6	SBB	10.26	56.89	78.16
	BZ	13.34	57.46	87.58
	YU	10.52	47.82	84.72
ANN2-SPH RF4	SBB	17.48	67.50	85.56
	BZ	18.49	72.96	96.57
	YU	20.34	75.21	96.65
ANN3-SPH RF3	SBB	20.39	84.10	97.45
	BZ	20.39	84.10	97.45
	YU	21.40	82.65	97.75

Table D.29 – X-Velocity Percent Error Results for the 3D ANN-SPH SGA Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-SPH SGA	SBB	18.18	50.65	66.23
	BZ	14.29	42.86	77.92
	YU	12.99	36.36	76.62
ANN2-SPH SGA	SBB	17.28	55.56	74.07
	BZ	17.28	67.90	88.89
	YU	19.14	67.90	89.51
ANN3-SPH SGA	SBB	15.27	61.09	77.09
	BZ	21.82	81.82	96.73
	YU	21.82	79.64	96.36

Table D.30 – Z-Velocity Percent Error Results for the 3D ANN-SPH Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-SPH RF6	SBB	14.53	43.33	63.98
	BZ	15.24	59.14	74.90
	YU	13.61	57.20	74.24
ANN2-SPH RF4	SBB	20.83	57.46	72.48
	BZ	13.61	65.70	81.81
	YU	15.37	66.58	81.68
ANN3-SPH RF3	SBB	12.81	66.93	82.39
	BZ	12.81	66.93	82.39
	YU	13.65	66.45	81.64

Table D.31 – Z-Velocity Percent Error Results for the 3D ANN-SPH SGA Cases

Grid	Boundary Treatment	%nodes with PE < 1%	%nodes with PE < 5%	%nodes with PE < 10%
ANN1-SPH SGA	SBB	2.67	23.53	40.64
	BZ	5.35	32.09	53.48
	YU	2.67	31.55	50.80
ANN2-SPH SGA	SBB	4.83	36.36	59.66
	BZ	6.53	42.33	67.33
	YU	6.53	44.32	67.33
ANN3-SPH SGA	SBB	11.01	48.43	72.20
	BZ	13.46	57.17	77.10
	YU	15.73	59.09	77.45

Force Evaluation on a Stationary Construct

Force and Moment Results for 3D Main Grid Cases

Table D.32 – Force Results for the 3D ANN-CUBE Main Grid Cases

Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000050		0.000561		0.00000811	
ANN1 CUBE	SBB	-0.000055	8.83%	0.000623	11.02%	0.00000701	13.48%
	BZ	-0.000044	11.62%	0.000698	24.31%	0.00000572	29.48%
	YU	-0.000041	18.36%	0.000582	3.65%	0.00001080	33.26%
ANN2 CUBE	SBB	-0.000052	3.65%	0.000549	2.16%	0.00001365	68.34%
	BZ	-0.000054	8.50%	0.000569	1.39%	0.00001466	80.91%
	YU	-0.000056	10.99%	0.000560	0.19%	0.00000685	15.49%
ANN3 CUBE	SBB	-0.000041	17.60%	0.000501	10.67%	0.00000449	44.60%
	BZ	-0.000033	33.35%	0.000506	9.93%	0.00000526	35.16%
	YU	-0.000040	21.24%	0.000544	3.14%	0.00000819	1.05%

Table D.33 – Force Results for the 3D ANN-SPH Main Grid Cases

Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000018		0.000353		0.00000306	
ANN1 SPH	SBB	-0.000022	23.12%	0.000381	8.00%	0.00000640	109.30%
	BZ	-0.000022	22.85%	0.000389	10.29%	0.00000743	142.91%
	YU	-0.000034	90.28%	0.000347	1.62%	0.00000190	37.84%
ANN2 SPH	SBB	-0.000019	3.88%	0.000351	0.66%	0.00000340	11.28%
	BZ	-0.000020	10.42%	0.000347	1.60%	0.00000386	26.22%
	YU	-0.000022	21.16%	0.000357	1.20%	0.00000344	12.43%
ANN3 SPH	SBB	-0.000018	1.53%	0.000344	2.41%	0.00000264	13.54%
	BZ	-0.000016	9.21%	0.000345	2.15%	0.00000269	12.14%
	YU	-0.000017	7.39%	0.000345	2.29%	0.00000361	18.21%

Table D.34 – Force Results for the 3D ANN-DSC Main Grid Cases

Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000026		0.000380		0.00000130	
ANN1 DSC	SBB	-0.000026	2.68%	0.000373	1.90%	0.00000324	148.65%
	BZ	-0.000022	16.27%	0.000353	7.14%	0.00000240	83.79%
	YU	-0.000029	9.52%	0.000396	4.19%	0.00000177	35.45%
ANN2 DSC	SBB	-0.000021	20.99%	0.000369	2.88%	0.00000137	4.95%
	BZ	-0.000013	50.97%	0.000346	8.96%	0.00000090	30.76%
	YU	-0.000022	14.66%	0.000373	1.90%	0.00000164	26.20%
ANN3 DSC	SBB	-0.000022	17.69%	0.000367	3.45%	0.00000228	74.99%
	BZ	-0.000015	40.92%	0.000371	2.37%	0.00000318	144.23%
	YU	-0.000019	28.39%	0.000377	0.97%	0.00000253	93.78%

Table D.35 – Force Results for the 3D ANN-EL Main Grid Cases

Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000024		0.000311		0.00000017	
ANN1 EL	SBB	-0.000024	2.04%	0.000314	1.13%	0.00000186	985.02%
	BZ	-0.000036	48.06%	0.000280	9.92%	0.00000297	1628.26%
	YU	-0.000047	96.79%	0.000310	0.08%	0.00000133	675.82%
ANN2 EL	SBB	-0.000022	8.89%	0.000303	2.38%	0.00000070	304.99%
	BZ	-0.000010	57.68%	0.000300	3.27%	0.00000173	908.20%
	YU	-0.000020	16.45%	0.000307	1.12%	0.00000101	489.74%
ANN3 EL	SBB	-0.000024	1.79%	0.000308	1.00%	-0.00000007	138.70%
	BZ	-0.000026	6.61%	0.000308	0.77%	-0.00000029	267.45%
	YU	-0.000022	6.35%	0.000319	2.84%	0.00000030	74.04%

Force and Moment Results for Sub-Grid Cases

Table D.36 – Force Results for the 3D ANN-CUBE Sub-Grid Cases

Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000050		0.000561		0.00000811	
ANN1 CUBE RF6	SBB	-0.000038	25.22%	0.000381	32.14%	0.00000847	4.46%
	BZ	-0.000038	24.75%	0.000456	18.70%	0.00000823	1.52%
	YU	-0.000037	25.61%	0.000451	19.66%	0.00000856	5.59%
ANN2 CUBE RF4	SBB	-0.000047	7.15%	0.000441	21.50%	0.00000754	6.98%
	BZ	-0.000040	20.35%	0.000467	16.81%	0.00000770	5.06%
	YU	-0.000041	17.43%	0.000479	14.60%	0.00000797	1.67%
ANN3 CUBE RF3	SBB	-0.000044	11.43%	0.000576	2.69%	0.00000887	9.44%
	BZ	-0.000042	16.52%	0.000536	4.49%	0.00000866	6.84%
	YU	-0.000041	18.60%	0.000578	3.01%	0.00000905	11.60%

Table D.37 – Force Results for the 3D ANN-SPH Sub-Grid Cases

Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000018		0.000353		0.00000306	
ANN1 SPH RF6	SBB	-0.000015	13.54%	0.000293	16.99%	0.00000282	7.73%
	BZ	-0.000013	25.08%	0.000302	14.36%	0.00000293	10.20%
	YU	-0.000014	23.09%	0.000302	14.45%	0.00000295	4.25%
ANN2 SPH RF4	SBB	-0.000016	9.24%	0.000299	15.24%	0.00000283	3.34%
	BZ	-0.000014	21.21%	0.000304	13.84%	0.00000287	6.08%
	YU	-0.000015	18.66%	0.000308	12.70%	0.00000292	4.54%
ANN3 SPH RF3	SBB	-0.000016	8.28%	0.000311	11.93%	0.00000282	7.86%
	BZ	-0.000015	16.62%	0.000316	10.38%	0.00000292	4.56%
	YU	-0.000015	15.68%	0.000318	9.96%	0.00000309	0.96%

Table D.38 – Force Results for the 3D ANN-DSC Sub-Grid Cases

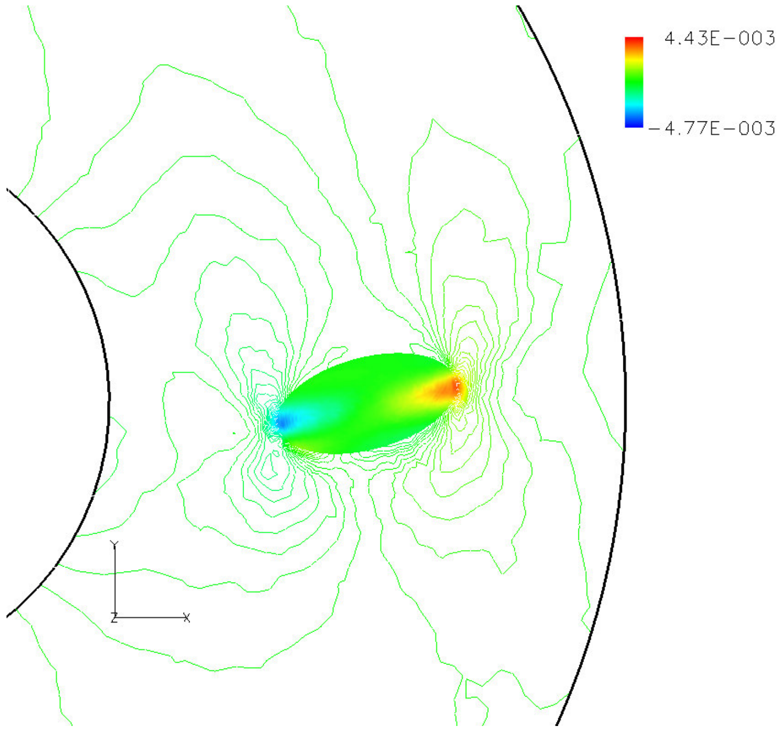
Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000026		0.000380		0.00000130	
ANN1	SBB	-0.000018	29.87%	0.000344	9.64%	0.00000131	0.29%
DSC	BZ	-0.000015	43.32%	0.000335	11.81%	0.00000154	17.99%
RF6	YU	-0.000015	43.91%	0.000342	10.21%	0.00000148	13.86%
ANN2	SBB	-0.000019	27.37%	0.000331	13.11%	0.00000143	10.09%
DSC	BZ	-0.000016	38.98%	0.000333	12.38%	0.00000155	19.19%
RF4	YU	-0.000016	38.16%	0.000340	10.53%	0.00000154	17.83%
ANN3	SBB	-0.000021	21.41%	0.000336	11.72%	0.00000130	0.00%
DSC	BZ	-0.000018	31.12%	0.000342	9.97%	0.00000147	12.63%
RF3	YU	-0.000017	33.52%	0.000349	8.19%	0.00000148	13.33%

Table D.39 – Force Results for the 3D ANN-EL Sub-Grid Cases

Grid	Boundary Treatment	X-Force (dN)	Error (%)	Y-Force (dN)	Error (%)	Z-Torque (dN-dm)	Error (%)
FLUENT		-0.000024		0.000311		0.00000017	
ANN1	SBB	-0.000021	13.67%	0.000279	10.19%	0.00000003	81.30%
EL	BZ	-0.000019	19.93%	0.000279	10.18%	0.00000002	90.66%
RF6	YU	-0.000020	15.90%	0.000283	8.84%	0.00000010	44.12%
ANN2	SBB	-0.000021	13.19%	0.000279	10.27%	0.00000007	56.35%
EL	BZ	-0.000019	19.69%	0.000274	11.65%	0.00000007	60.98%
RF4	YU	-0.000021	12.81%	0.000281	9.39%	0.00000015	9.94%
ANN3	SBB	-0.000021	13.39%	0.000278	10.64%	0.00000017	0.88%
EL	BZ	-0.000020	16.92%	0.000284	8.71%	0.00000011	38.48%
RF3	YU	-0.000020	15.85%	0.000288	7.20%	0.00000024	38.51%

Shear Stress Evaluation for Flow in an Annulus

<u>Contour Plots</u>



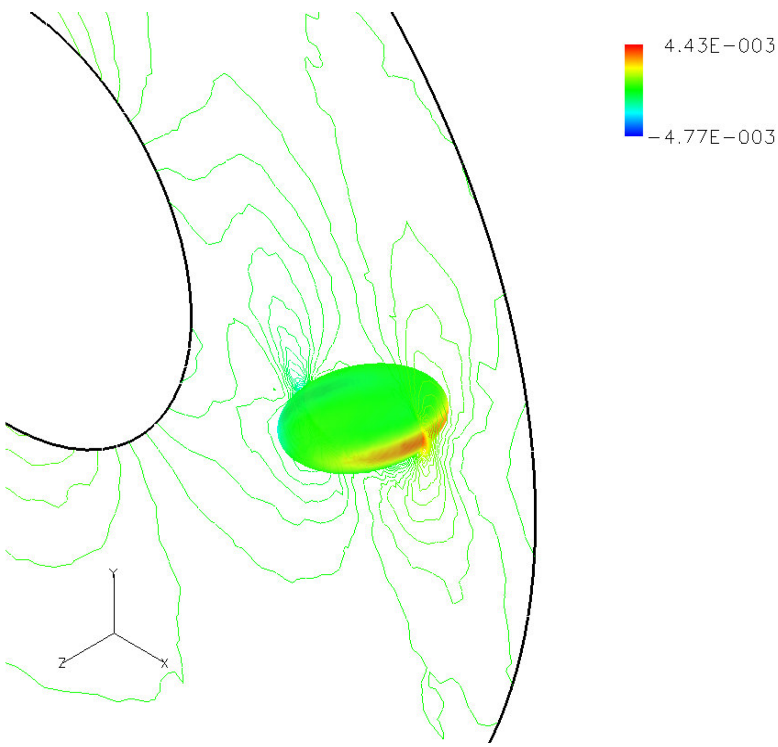


Figure D.57 – FLUENT Shear Stress Contour Plots for the ANN-EL Construct Case

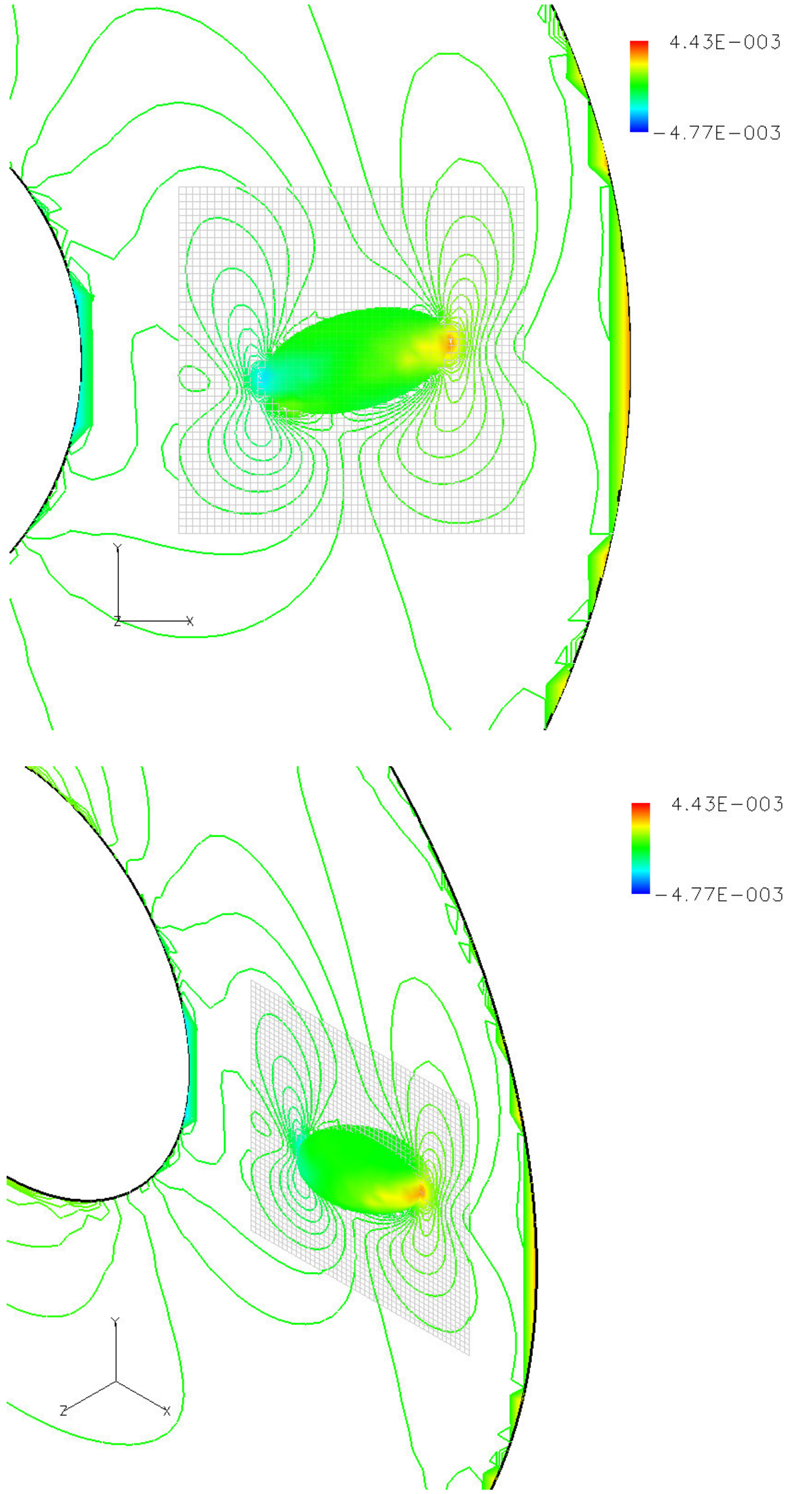


Figure D.58 – LB Shear Stress Contour Plots for the ANN-EL Construct Case

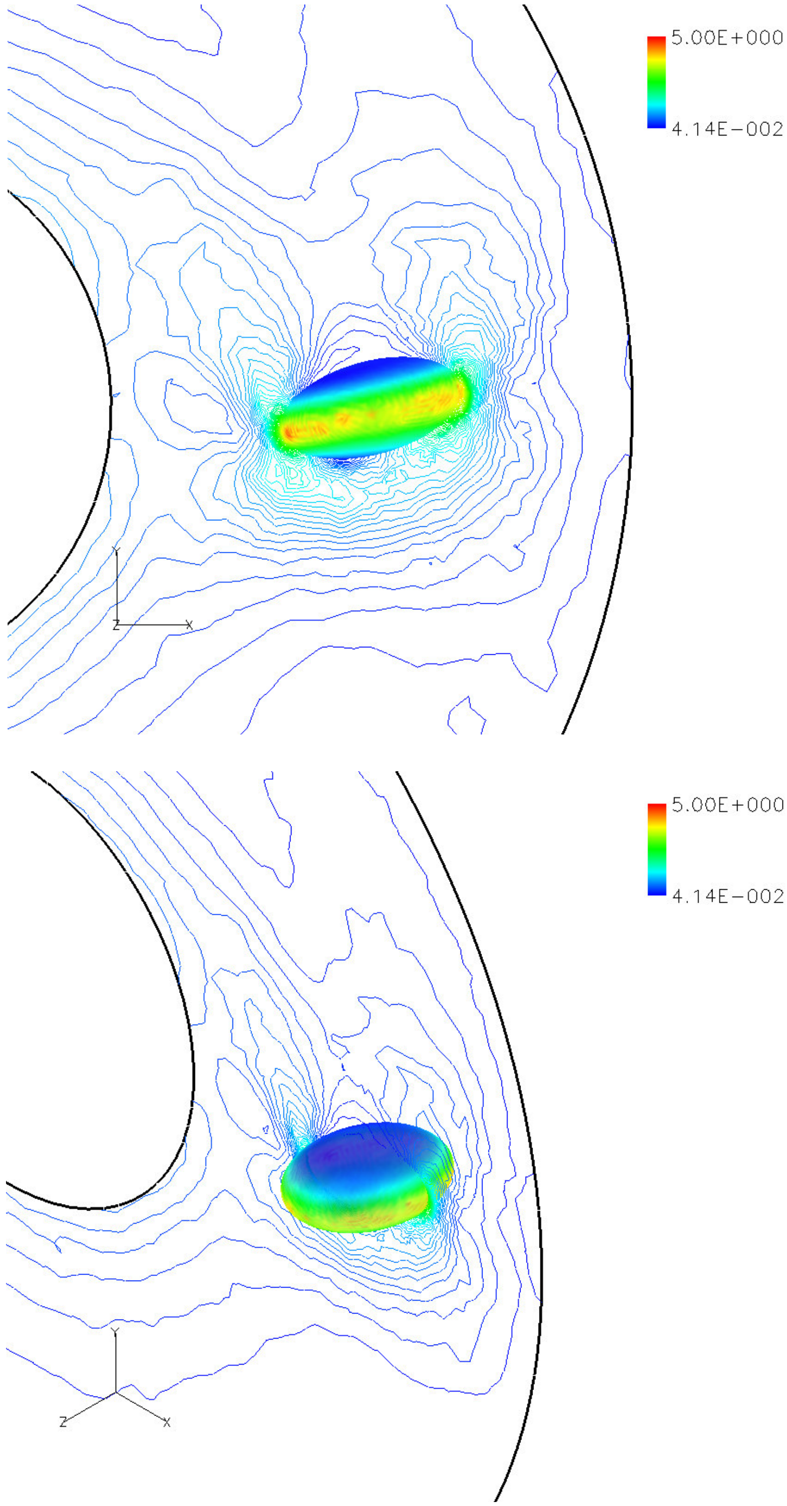


Figure D.59 – FLUENT Shear Rate Contour Plots for the ANN-EL Construct Case

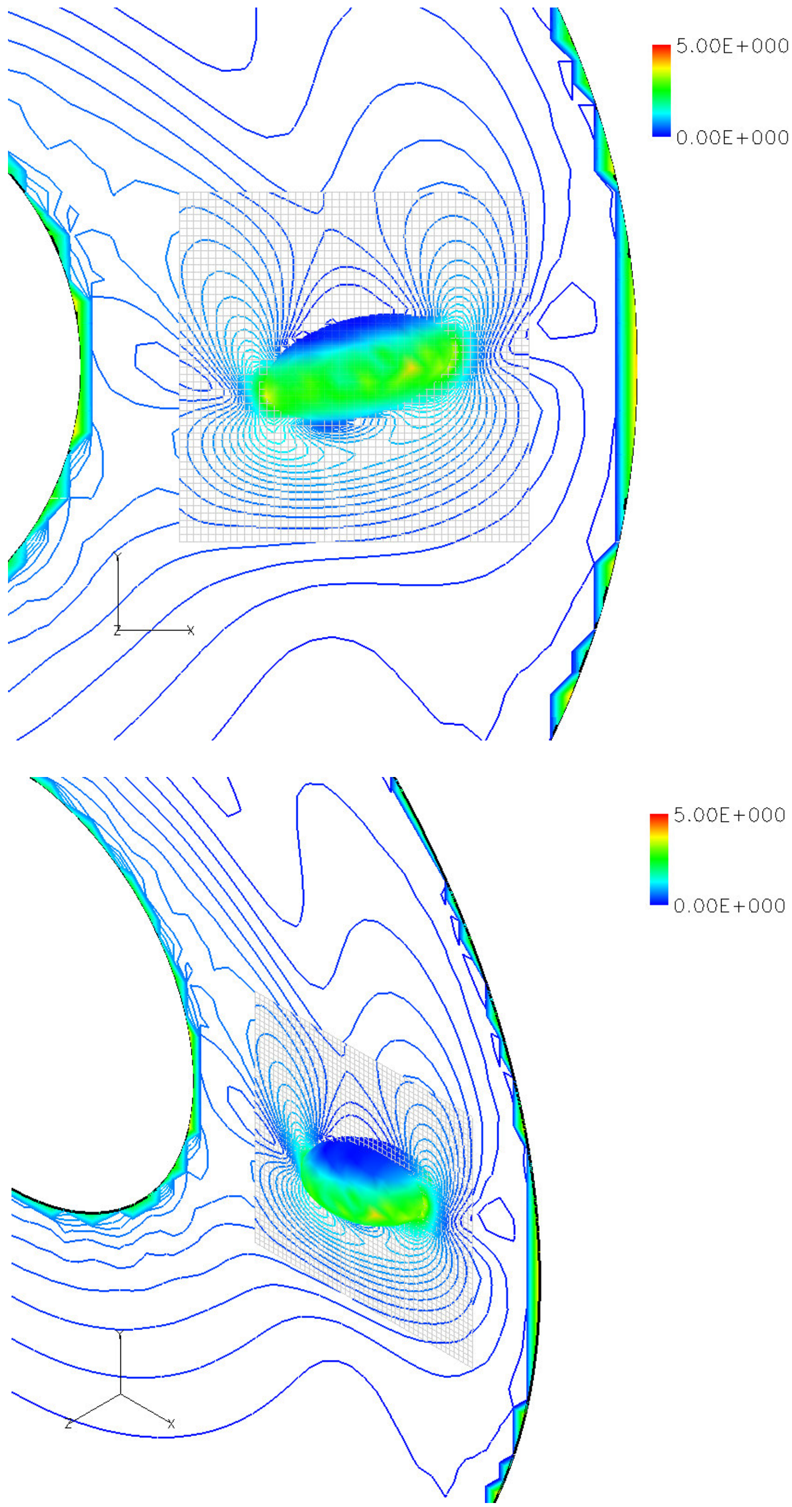


Figure D.60 – LB Shear Rate Contour Plots for the ANN-EL Construct Case

Shear Stress and Shear Rate Line Plots

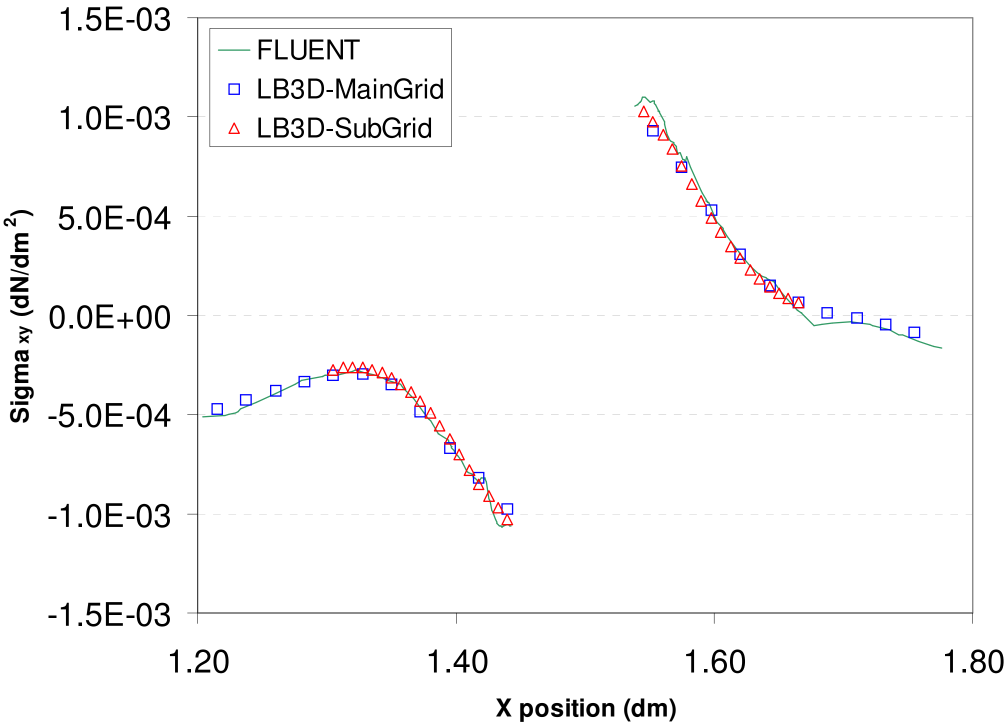


Figure D.61 – XY Shear Stress Plot Comparing Main Grid and Sub-Grid Values

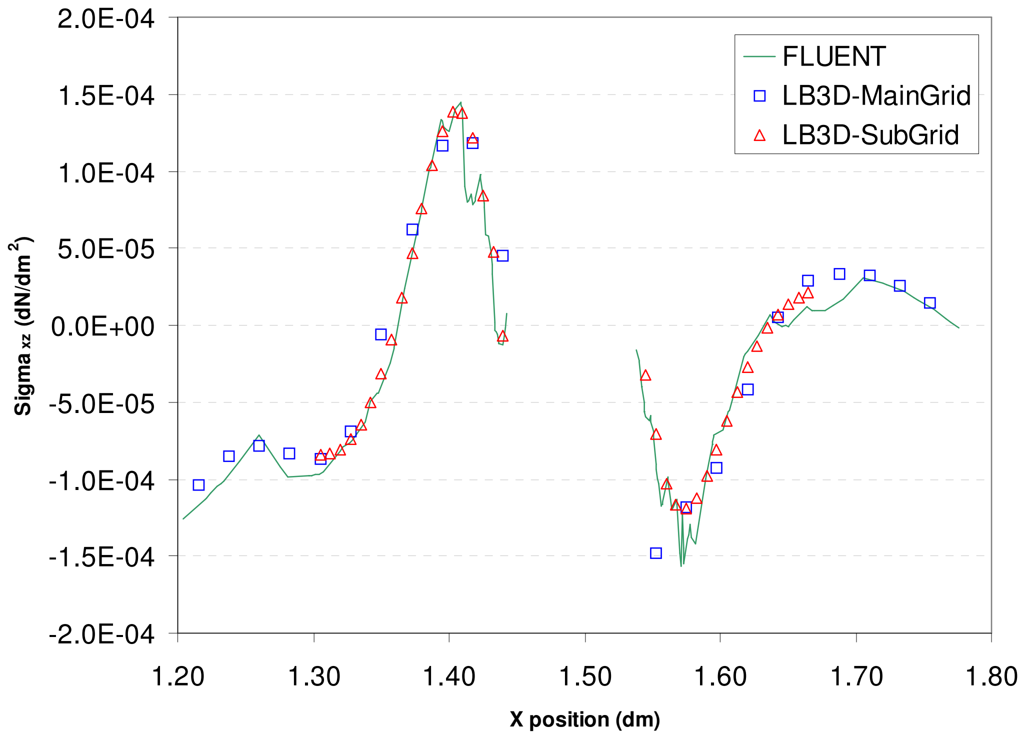


Figure D.62 – XZ Shear Stress Plot Comparing Main Grid and Sub-Grid Values

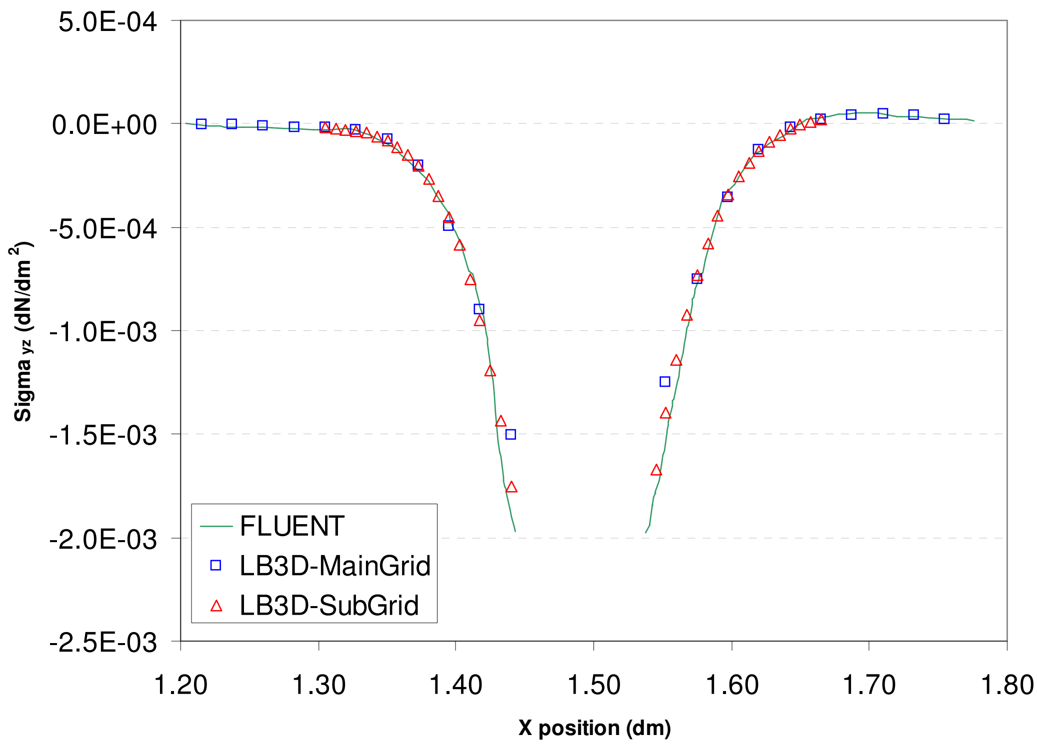


Figure D.63 – YZ Shear Stress Plot Comparing Main Grid and Sub-Grid Values

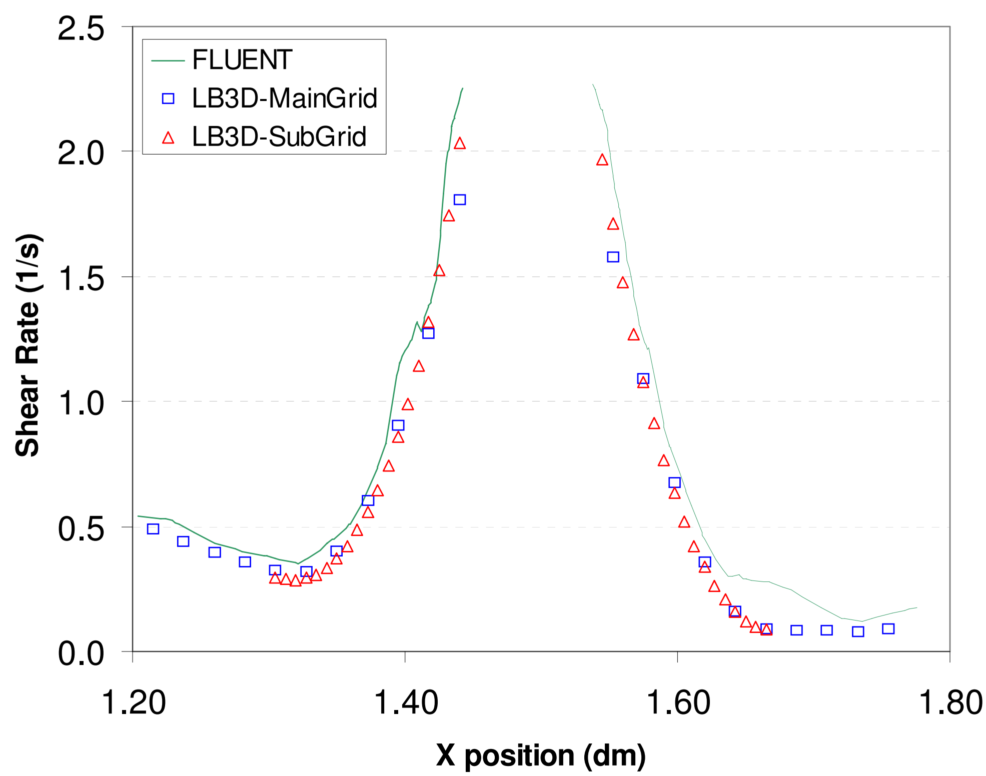


Figure D.64 – Shear Rate Plot Comparing Main Grid and Sub-Grid Values

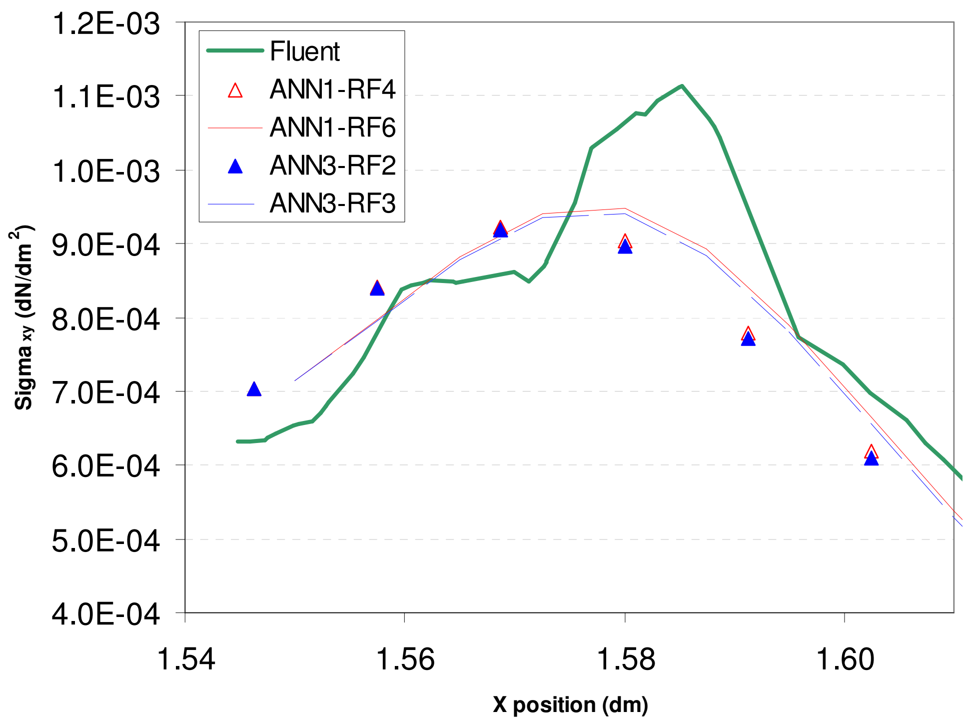


Figure D.65 – XY Shear Stress Plot Comparing Results across Main Grids

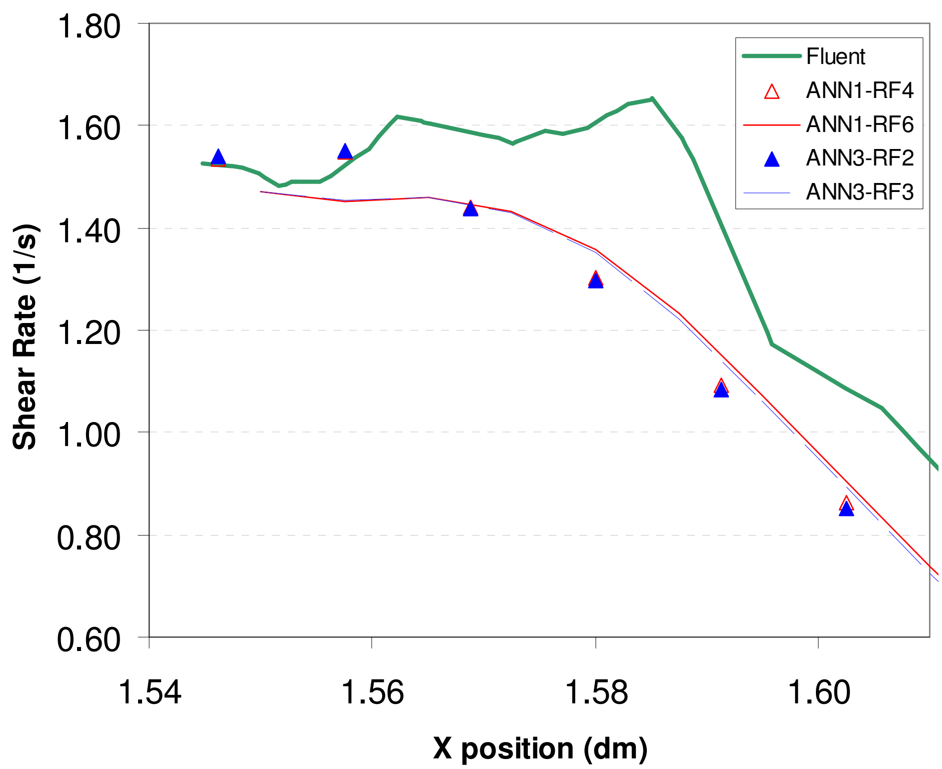


Figure D.66 – Shear Rate Plot Comparing Results across Main Grids

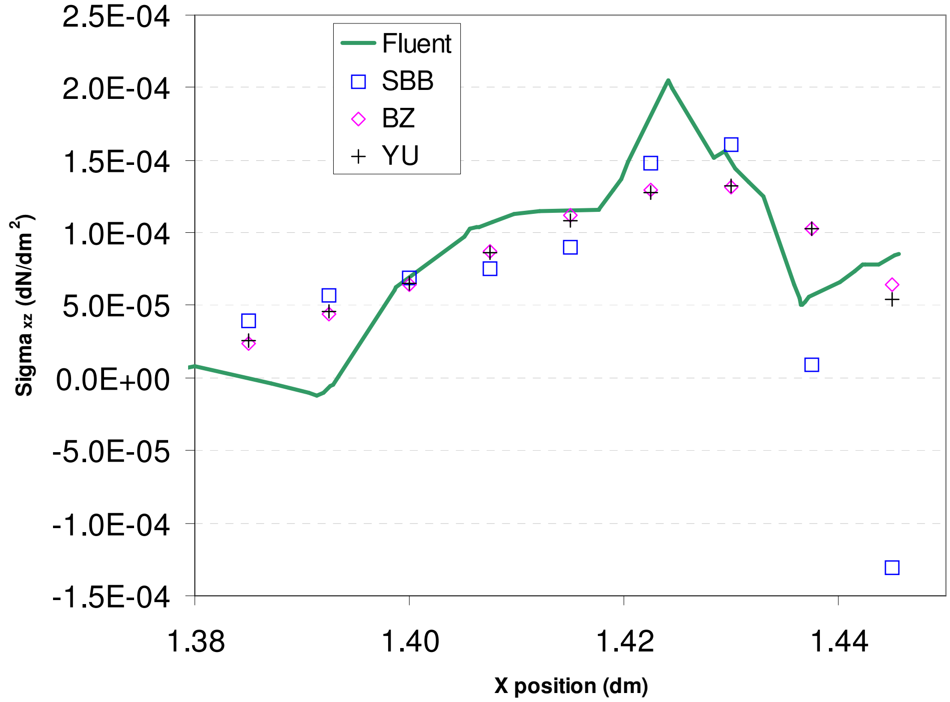


Figure D.67 – XZ Shear Stress Plot Comparing Results across Boundary Treatments

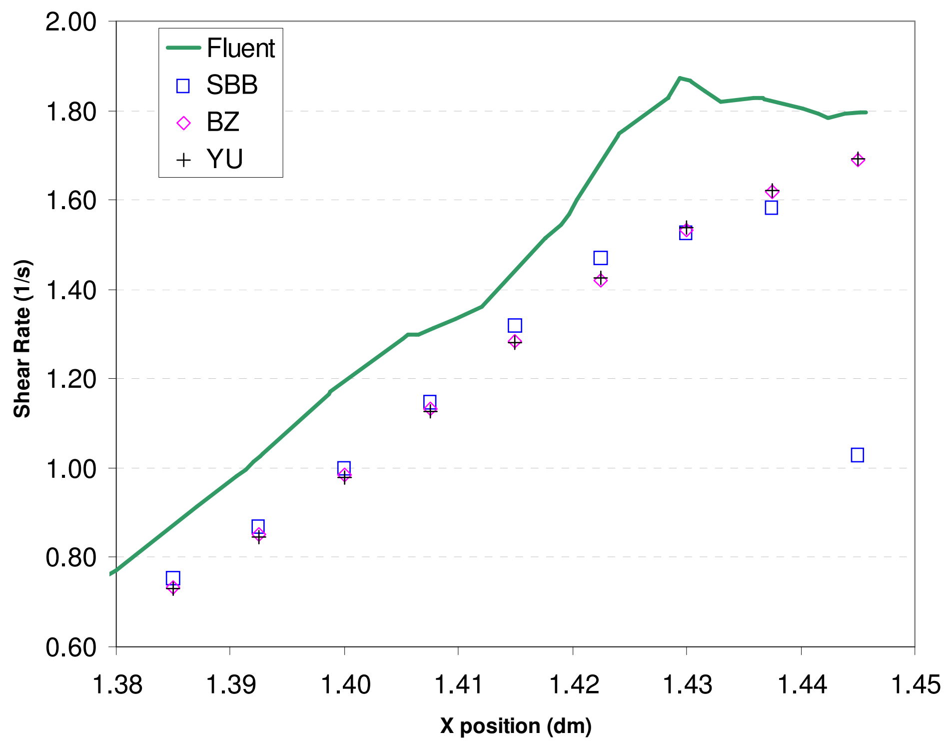


Figure D.68 – Shear Rate Plot Comparing Results across Boundary Treatments

Non-Dimensional Force versus Gap Size Plots

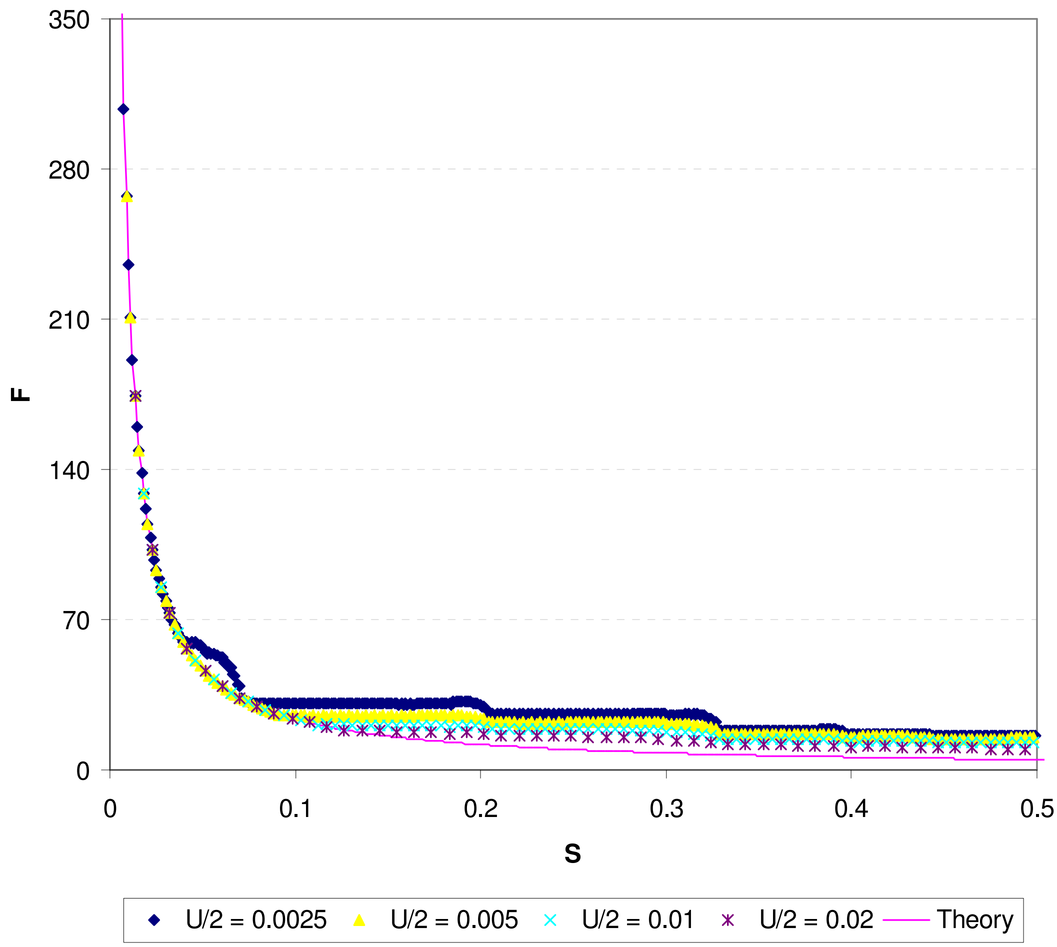


Figure D.69 – Force versus Gap Size for the SBB Boundary Treatment

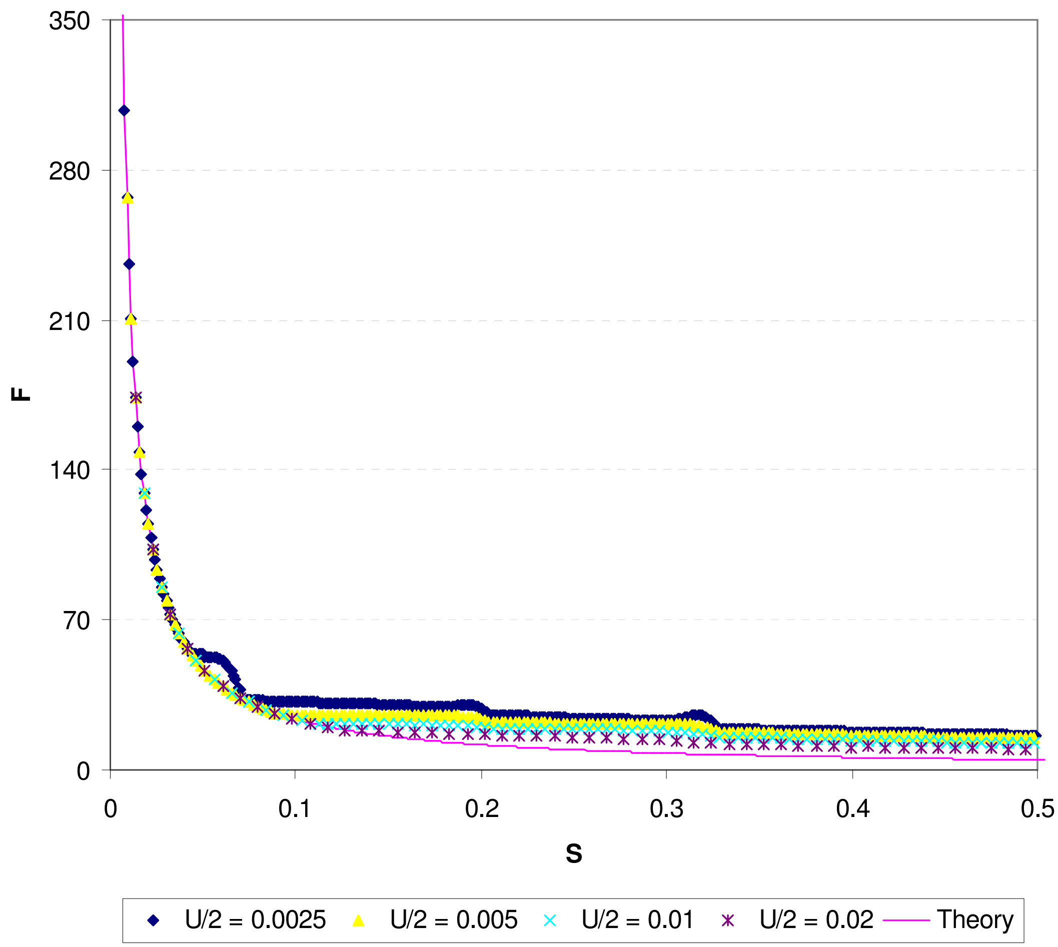


Figure D.70 – Force versus Gap Size for the BZ Boundary Treatment

Drag Force versus Channel Geometry Plots

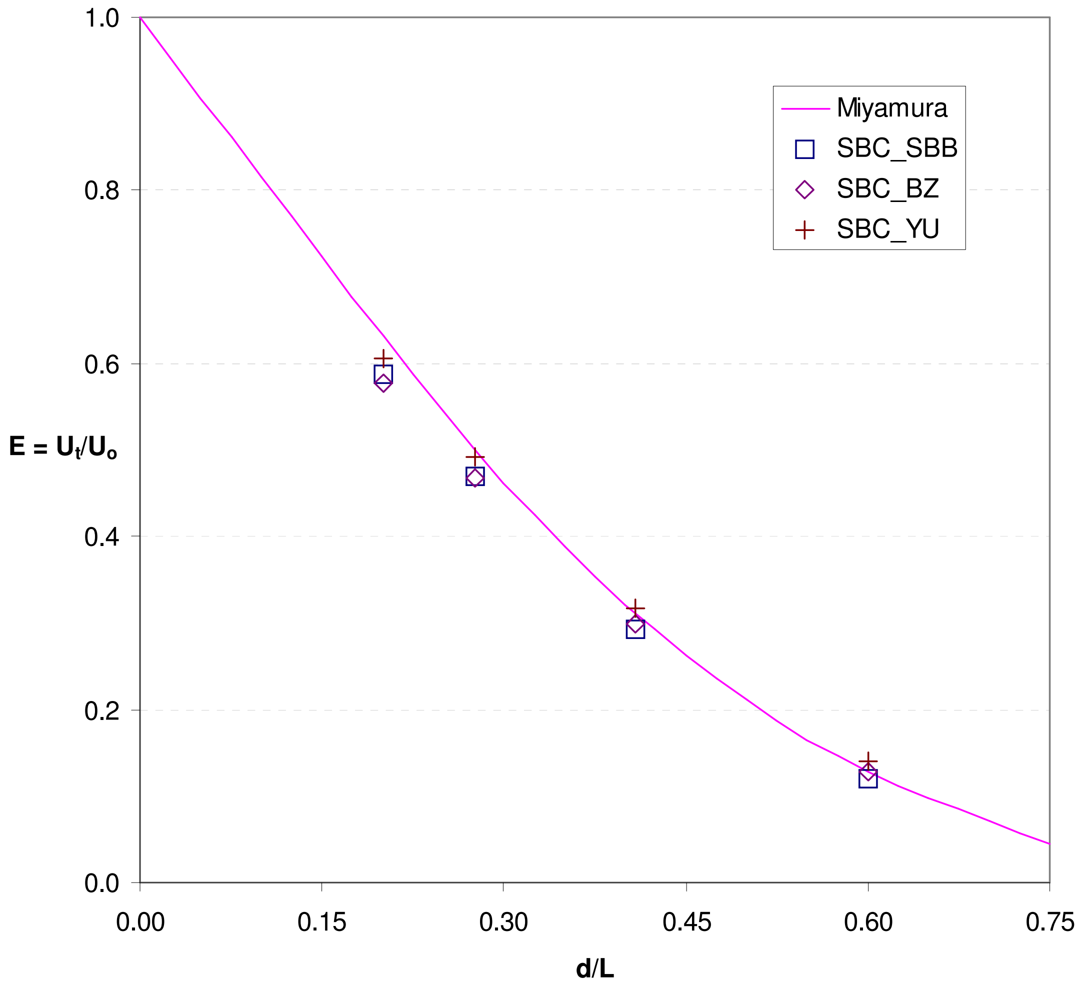


Figure D.71 – E versus d/L for all Boundary Treatments

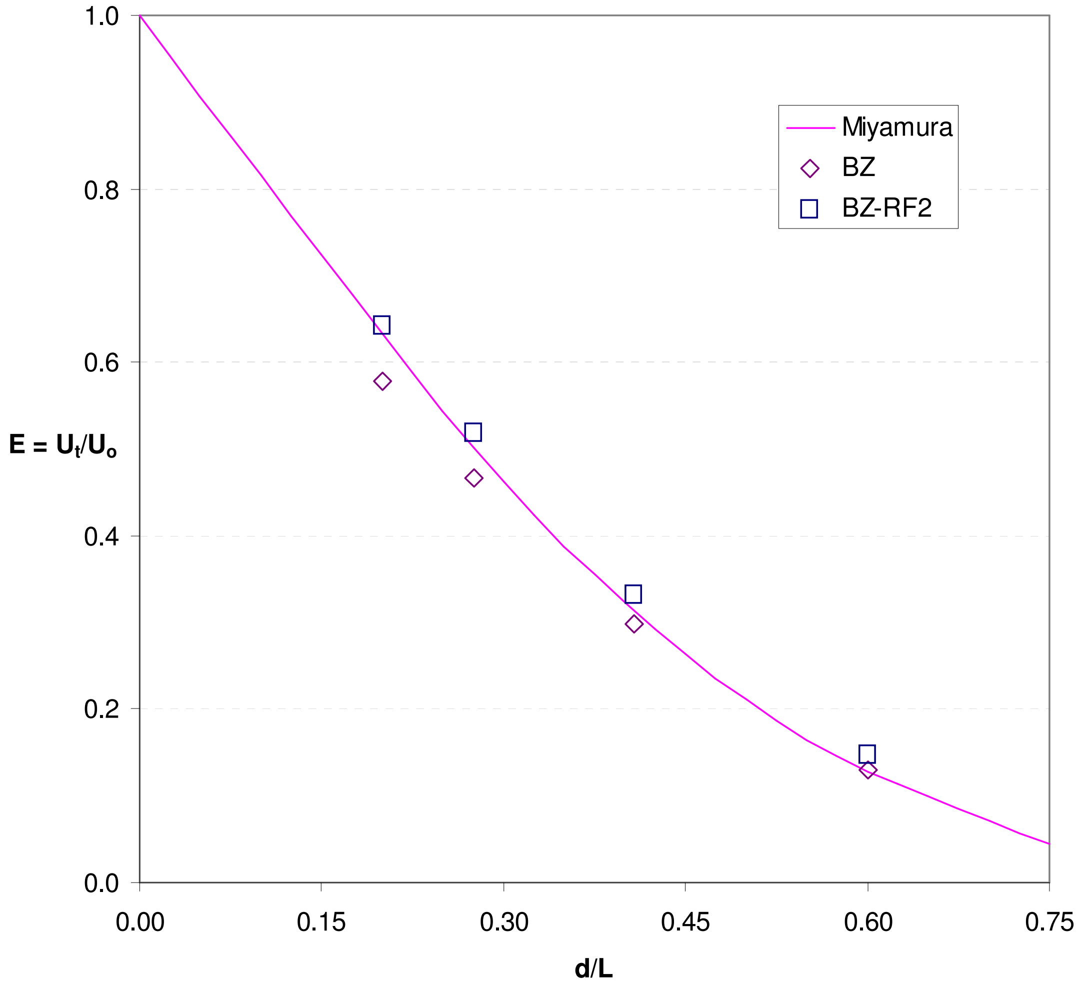


Figure D.72 – E versus d/L with Grid Refinement for the BZ Boundary Treatment

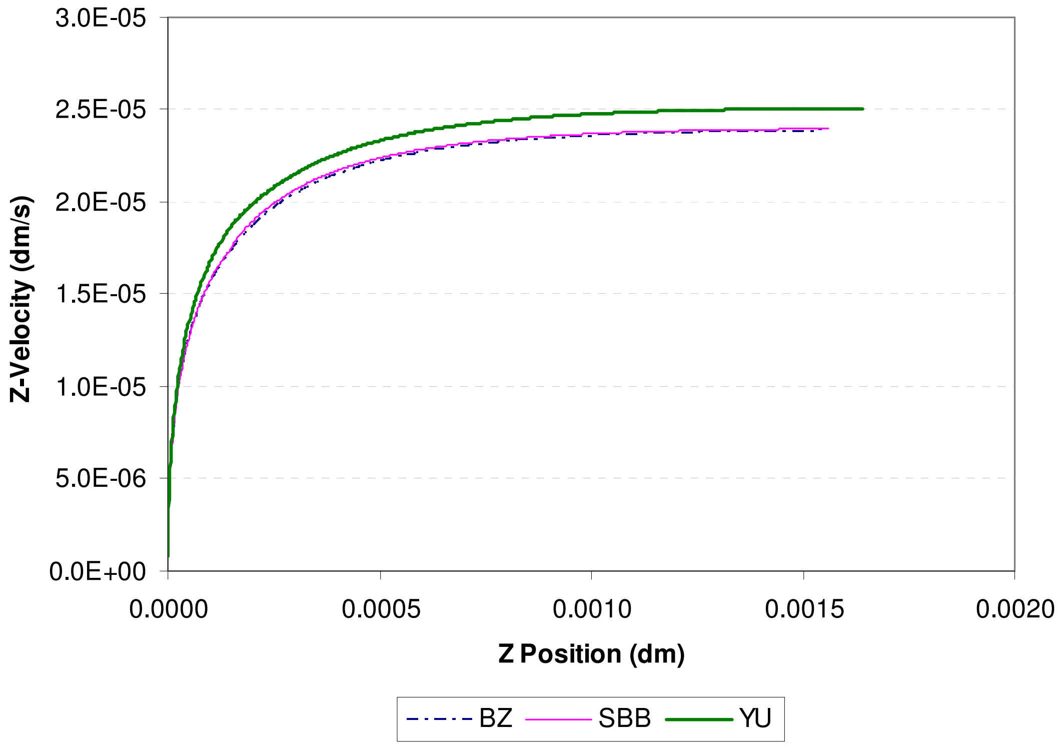


Figure D.73 – Particle Velocity versus Position for the 3D VCF3 Grid

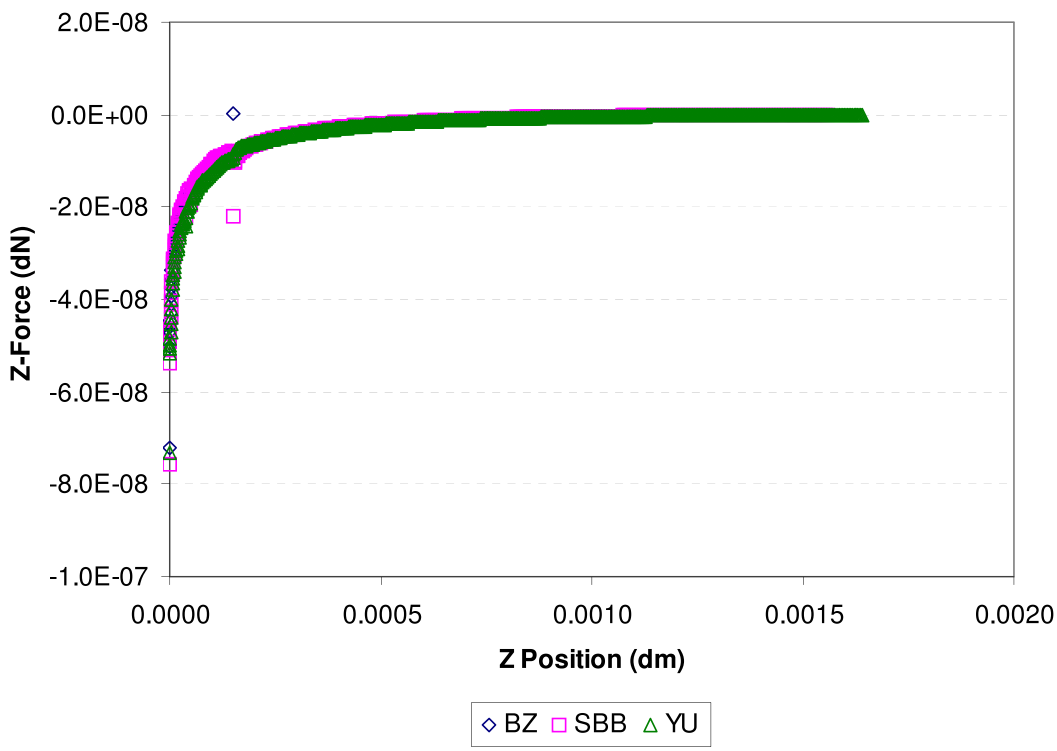


Figure D.74 – Particle Force versus Position for the 3D VCF3 Grid

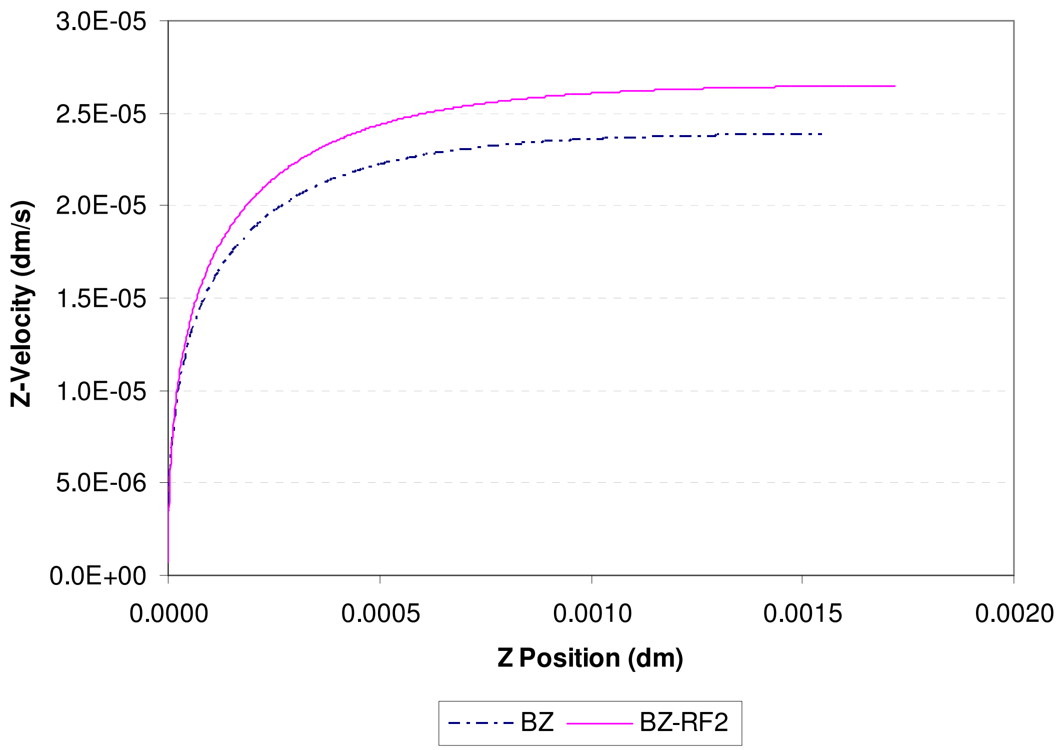


Figure D.75 – Particle Velocity versus Position for the 3D VCF3-RF Grids

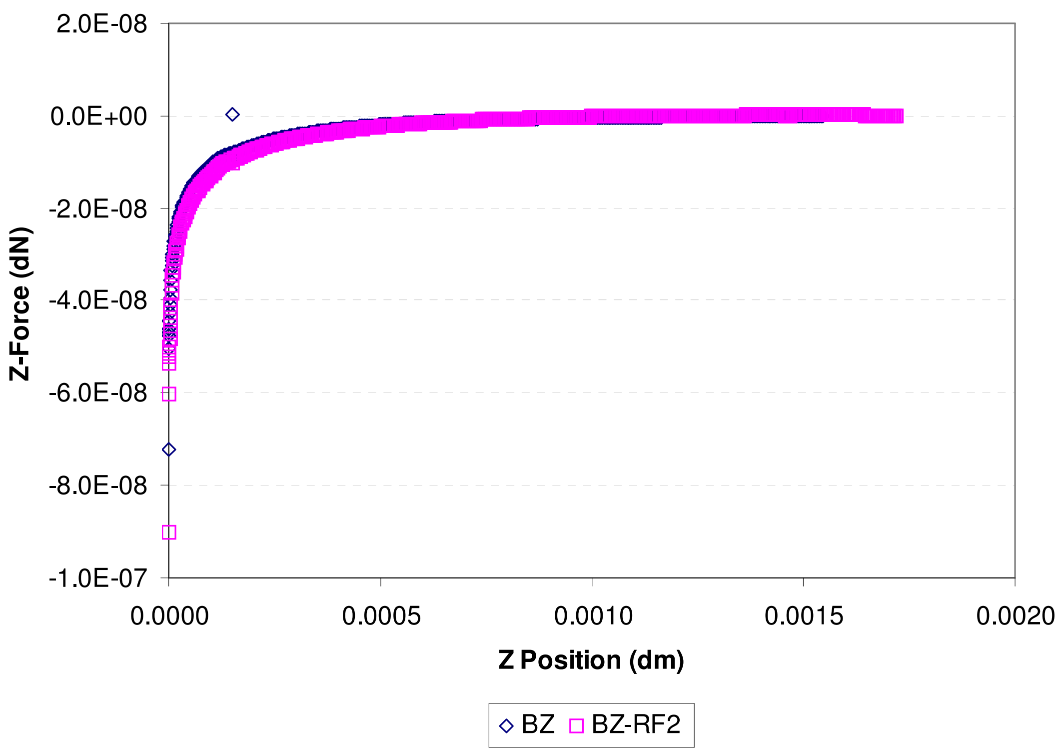


Figure D.76 – Particle Force versus Position for the 3D VCF3-RF Grid

Motion of a Neutrally-Buoyant Particle in a 3D Annulus

Particle Path Plots

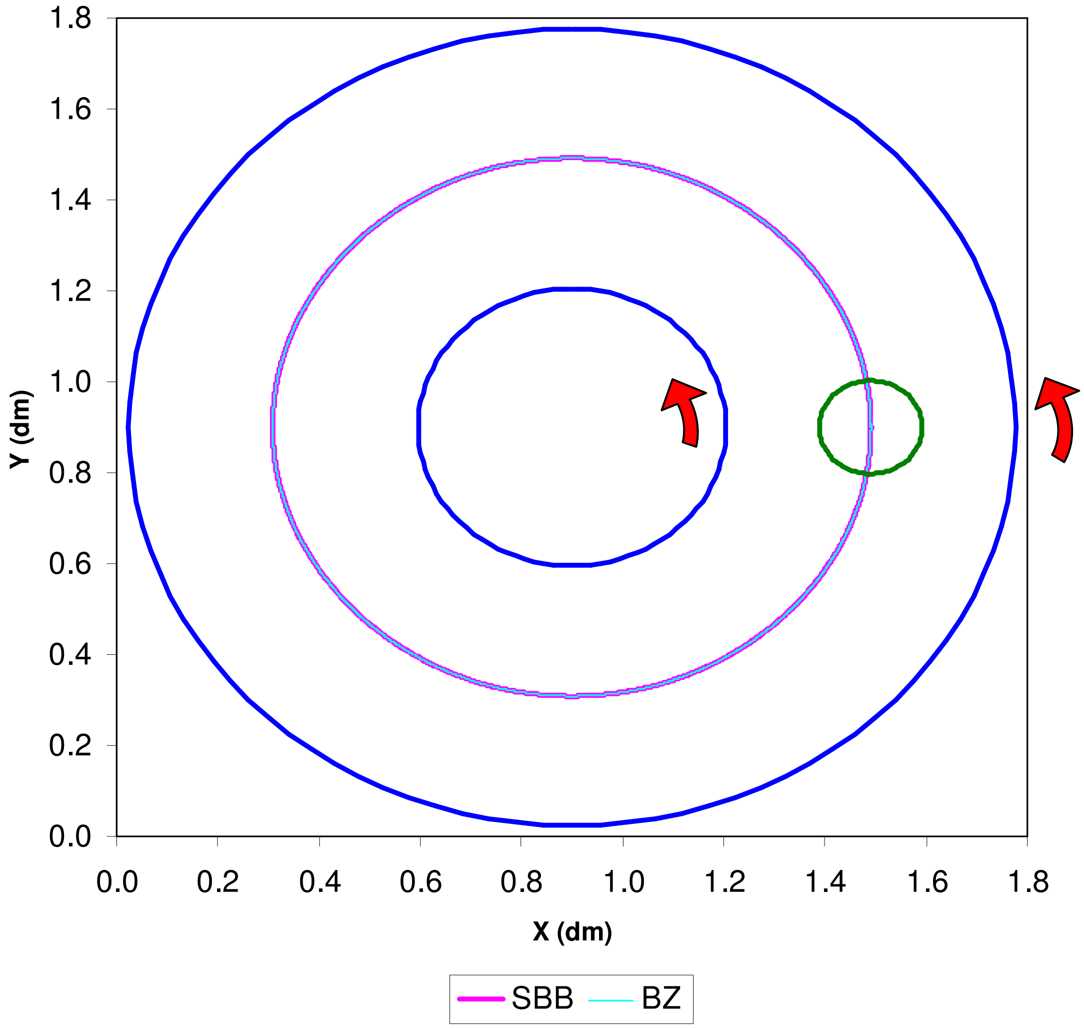


Figure D.77 – Sphere Particle Path for 3D ANN3 Case

Velocity Vector Visualization

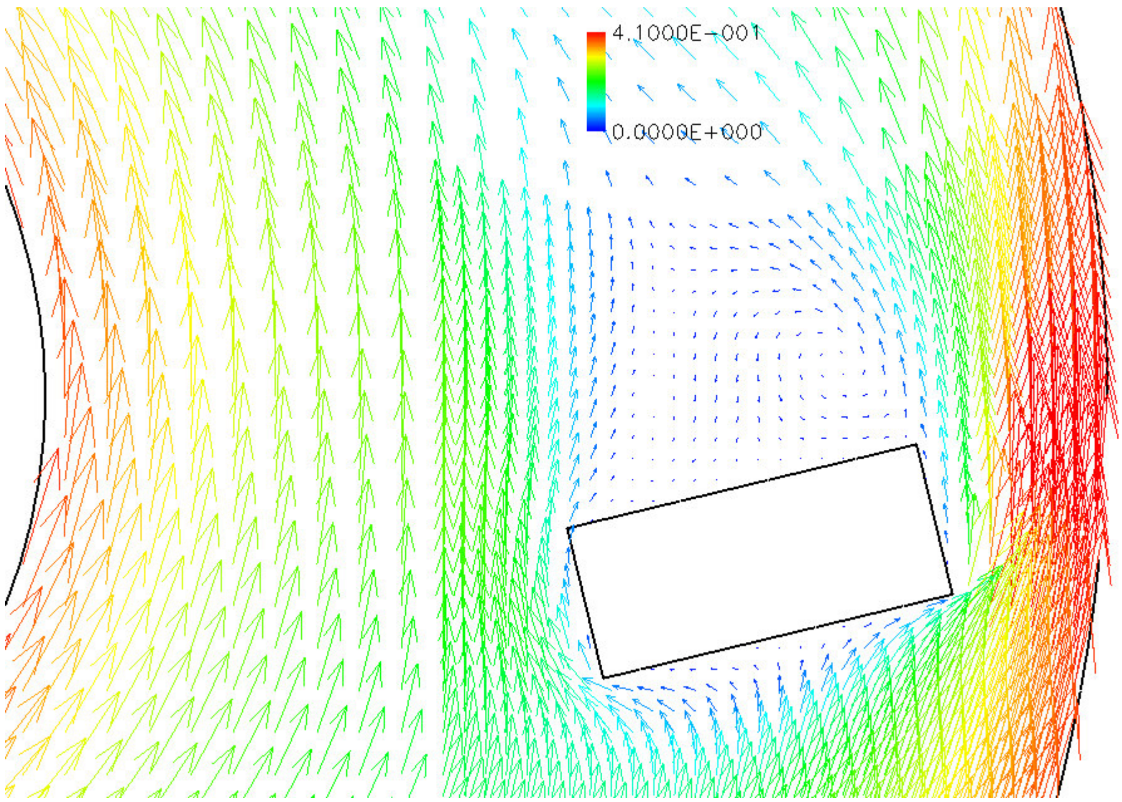


Figure D.78 – Velocity Vectors for 3D Flow around a DSC Construct

Particle Motion Paths

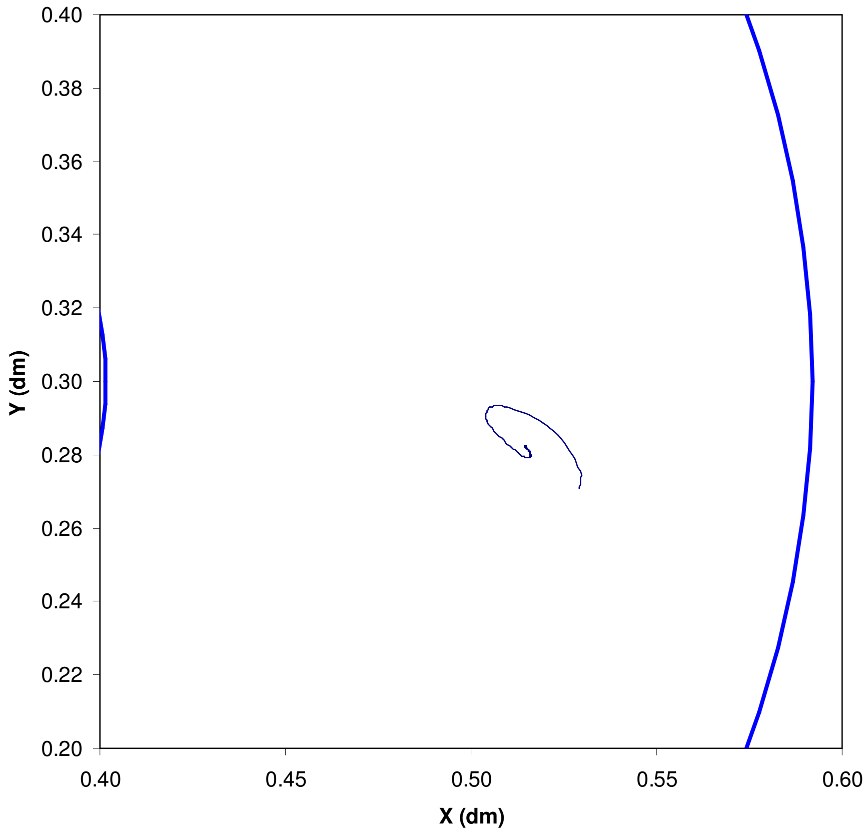


Figure D.79 – DSC Construct Path for Stationary Case

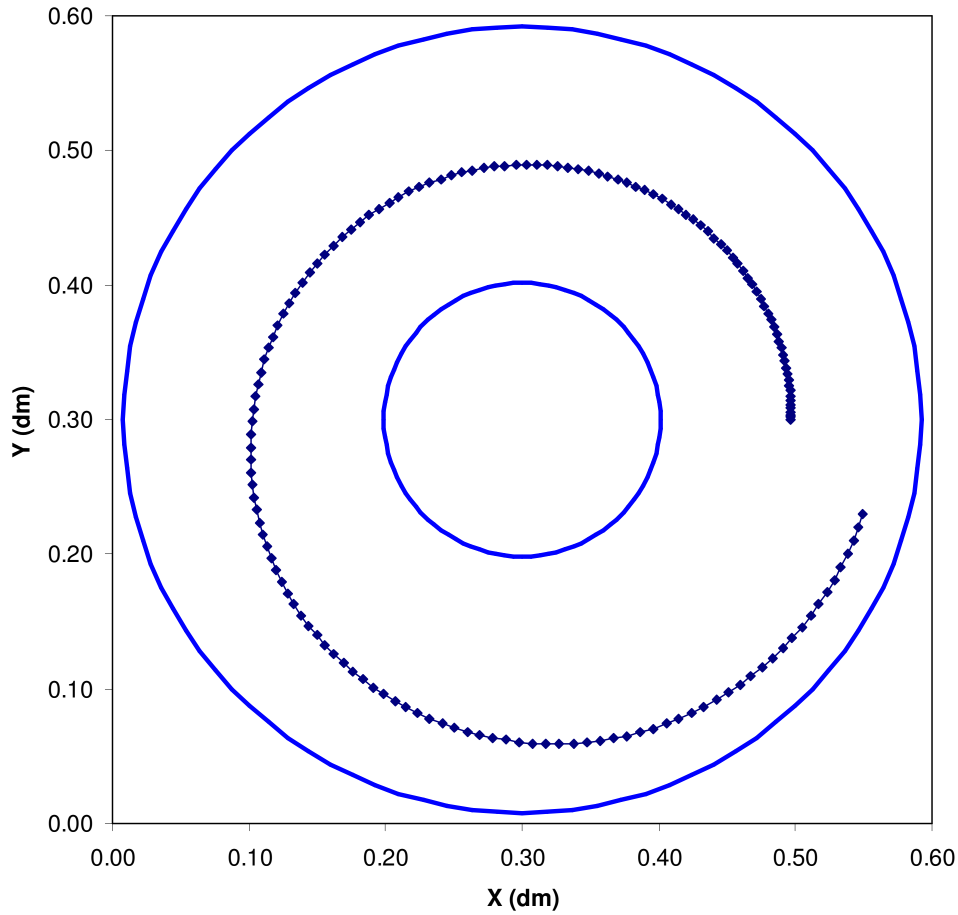


Figure D.80 – DSC Construct Path for Orbiting Annulus Case

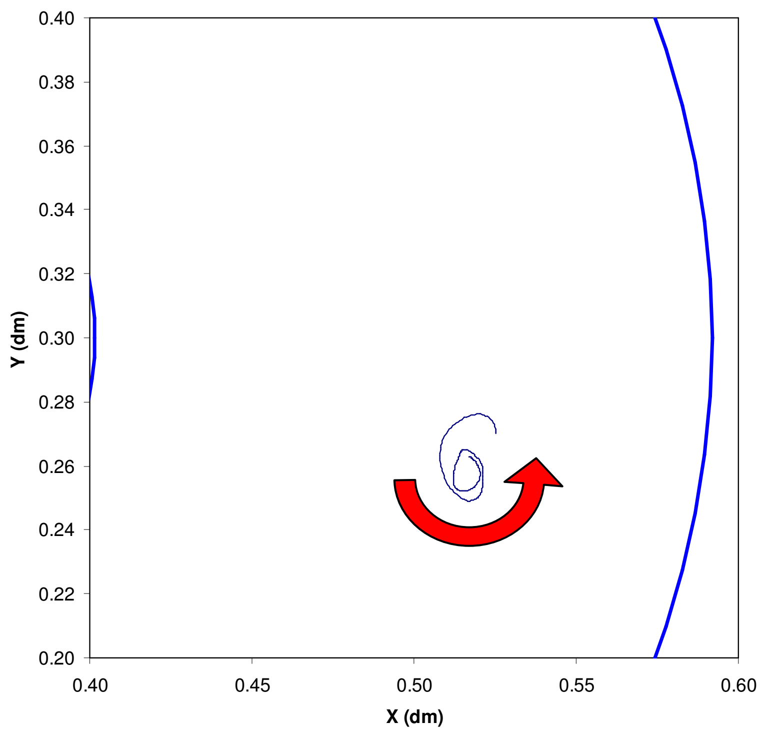


Figure D.81 – DSC Construct Path for the 13RPM Case

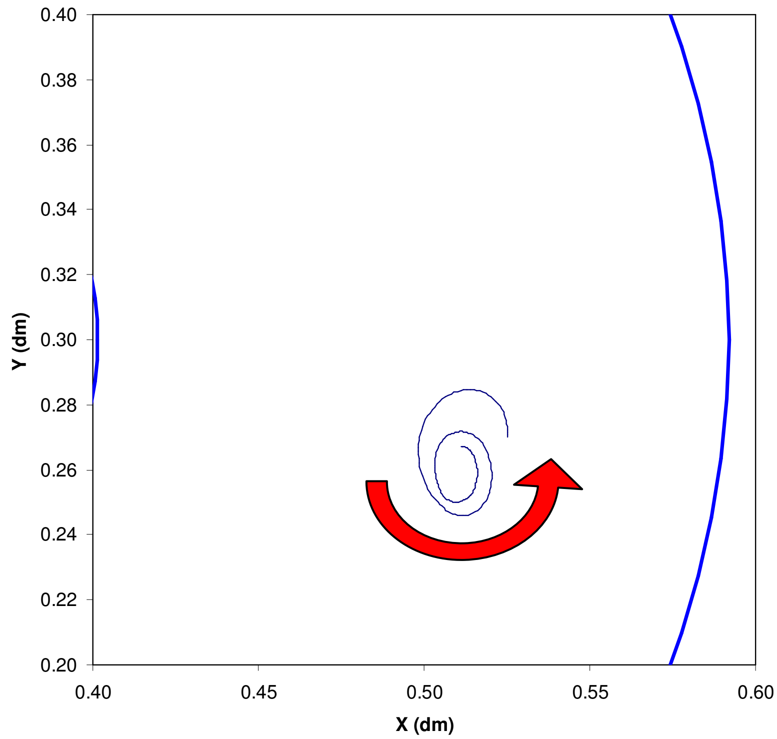


Figure D.82 – DSC Construct Path for the 14RPM Case

Shear Stress Contour Plots

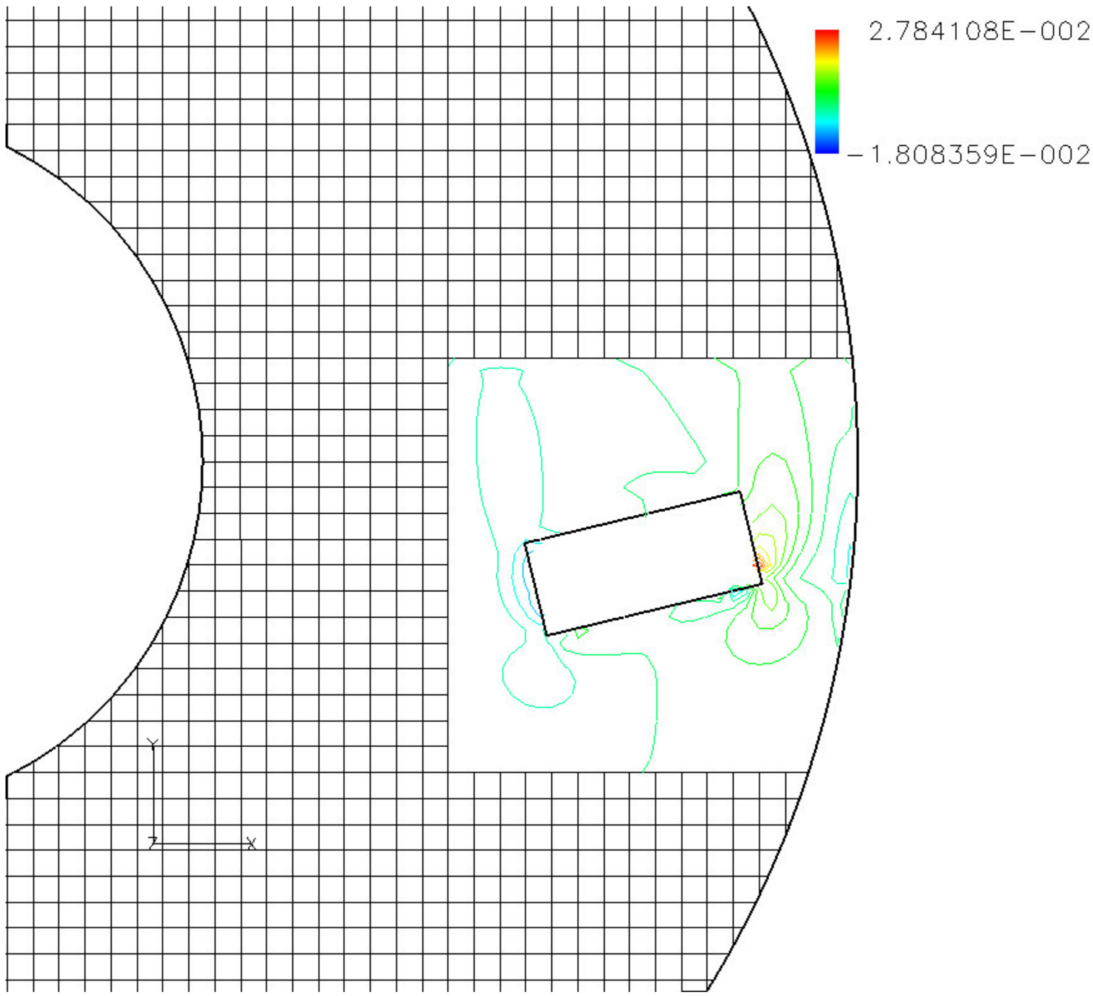


Figure D.83 – XY Shear Stress Contours for the DSC Construct

References

AIAA. Guide for the Verification and Validation of Computational Fluid Dynamics Simulations. 1998. AIAA G-077-1998.

Aidun CK, and Lu Y. Lattice Boltzmann Simulation of Solid Particles Suspended in Fluid. *J Stat Phys*. 1995. 81:49-61.

Aidun CK, Lu Y, and Ding EJ. Direct Analysis of Particulate Suspensions with Inertia using the Discrete Boltzmann Equation. *J Fluid Mech*. 1998. 373:287-311.

Aidun CK, and Ding EJ. Dynamics of Particle Sedimentation in a Vertical Channel: Period-Doubling Bifurcation and Chaotic State. *Phys Fluids*. 2003. 15(6):1612-1621.

Batchelor GK. An Introduction to Fluid Dynamics. 1967. Cambridge University Press, Cambridge, UK.

Begley CM, and Kleis SJ. The Fluid Dynamic and Shear Environment in the NASA/JSC Rotating Wall Perfused-Vessel Bioreactor. *Biotech Bioeng*. 2000. 70(1):32-40.

Benzi R, Succi S, and Vergassola M. The Lattice Boltzmann Equation: Theory and Applications. *Physics Reports*. 1992. 222(3):145-197.

Bhatnagar PL, Gross EP, and Krook M. A Model for Collision Processes in Gases. I: Small Amplitude Processes in Charged and Neutral One-Component Systems. *Phys Rev*. 1954. 94:511-525.

Botchwey EA, Pollack SR, Levine EM, and Laurencin CT. Bone Tissue Engineering in a Rotating Bioreactor Using a Microcarrier Matrix System. *J Biomed Mat Res*. 2001. 55(2):242-253.

Bouzidi M, Firdaouss M, and Lallemand P. Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Phys Fluids*. 2001. 13(11):3453-3459.

Brown JB. An Experimental Facility for the Investigation of Flow in a Circular-Couette Flow Bioreactor. Thesis-School of Mechanical Engineering, Georgia Institute of Technology. 1998.

Buick JM, and Greated CA. Gravity in a Lattice Boltzmann Model. *Phys Rev E*. 2000. 61(5):5307-20.

Cao N, Chen S, Jin S, and Martinez D. Physical Symmetry and Lattice Symmetry in the Lattice Boltzmann Method. *Phys Rev E*. 1997. 55(1):R21-R24.

ten Cate A, Nieuwstad CH, Derksen JJ, Van den Akker HEA. Particle Imaging Velocimetry Experiments and Lattice-Boltzmann Simulations on a Single Sphere Settling under Gravity. *Phys Fluids*. 2002. 14(11):4012-4025.

Chen S, and Doolen GD. Lattice Boltzmann Method for Fluid Flows. *Annu Rev Fluid Mech*. 1998. 30:329-64.

Chen S, Martinez D, and Mei R. On Boundary Conditions in Lattice Boltzmann Methods. *Phys Fluids*. 1996. 8(9):2527-2536.

Chen S, Wang Z, Shan XW, and Doolen GD. Lattice Boltzmann Computational Fluid Dynamics in Three Dimensions. *J Stat Phys*. 1992. 68:379-400.

Das P, Schurman DJ, Smith RL. Nitric Oxide and G Proteins Mediate the Response of Bovine Articular Chondrocytes to Fluid-Induced Shear. *J Orthop Res*. 1997. 15(1):87-93.

Dewey CF Jr, Bussolari SR, Gimbrone MA Jr, Davies PF. The Dynamic Response of Vascular Endothelial Cells to Fluid Shear Stress. *J Biomech Eng*. 1981. 103(3):177-85.

Ding EJ, and Aidun CK. Extension of the Lattice-Boltzmann Method for Direct Simulation of Suspended Particles Near Contact. *J Stat Phys*. 2003. 112:685-708.

Dooling D. Bioreactor Expands Health Research. http://science.msfc.nasa.gov/newhome/br/bioreactor.htm.

Faxen OH. Forces Exerted on a Rigid Cylinder in a Viscous Fluid Between Two Parallel Fixed Planes. *Proc R Swed Acad Eng Sci.* 1946. 187.

Feng J, Hu HH, and Joseph DD. Direct Simulation of Initial Value Problems for the Motion of Solid Bodies in a Newtonian Fluid Part 1. Sedimentation. *J Fluid Mech.* 1994. 261:95-134.

Feng ZG, and Michaelides EE. Interparticle Forces and Lift on a Particle Attached to a Solid Boundary in Suspension Flow. *Phys Fluids*. 2002. 14(1):49-60.

Filippova O, and Hanel D. Grid Refinement for Lattice-BGK Models. *J Comp Phys*. 1998a. 147(1):219-228.

Filippova O, and Hanel D. Boundary-Fitting and Local Grid Refinement for Lattice-BGK Models. *Int J Mod Phys C*. 1998b. 9(8):1271-1279.

Filippova O, and Hanel D. Acceleration of Lattice-BGK Schemes with Grid Refinement. *J Comp Phys*. 2000. 165:407-427.

FLUENT Version 6.1 User's Guide. 2003.

Freed D. Lattice-Boltzmann Method for Macroscopic Porous Media Modeling. *Int J of Mod Phys C*. 1998. 9(8):1491-1503.

Freed LE, Hollander AP, Martin I, Barry JR, Langer R, Vunjak-Novakovic G. Chondrogenesis in a Cell-Polymer-Bioreactor System. *Exp Cell Res*. 1998. 63(2):197-205.

Freed LE, Vunjak-Novakovic G, Langer R. Cultivation of Cell-Polymer Cartilage Implants in Bioreactors. *J Cell Biochem*. 1993. 51(3):257-64.

Freed LE, Vunjak-Novakovic G. Cultivation of Cell-Polymer Tissue Constructs in Simulated Microgravity. *Biotechnol Bioeng*. 1995. 46(4):306-13.

Frisch U, Hasslacher B, and Pomeau Y. Lattice-Gas Automata for the Navier-Stokes Equations. *Phys Rev Lett*. 1986. 56:1505-8

Frisch U, d'Humieries D, Hasslacher B, Lallemand P, Pomeau Y, and Rivet JP. Lattice Gas Hydrodynamics in Two and Three Dimensions. *Complex Systems*. 1987. 1:649-707.

Gallivan MA, Noble DR, Georgiadis JG, and Buckius RO. An Evaluation of the Bounce-Back Boundary Condition for Lattice Boltzmann Simulations. *Int J for Num Methods in Fluids*. 1997. 25:249-263.

Geist A, Beguelin A, Dongarra J, Jiang W, Manchek R, and Sunderam V. PVM: Parallel Virtual Machine – A Users' Guide and Tutorial for Networked Parallel Computing. 1994. MIT Press, Cambridge, MA.

Glowinski R, Pan TW, Helsa TI, Joseph DD, and Periaux J. A Fictitious Domain Approach to the Direct Numerical Simulation of Incompressible Viscous Flow Past Moving Rigid Bodies, Application to Particulate Flow. *J Comp Phys*. 2001. 169:302-362.

Grunau D, Chen S, and Eggert K. A Lattice Boltzmann Model for Multiphase Fluid Flows. *Phys Fluids A*. 1993. 5:2557-62.

He X, and Doolen GD. Lattice Boltzmann Method on Curvilinear Coordinates System: Flow Around a Circular Cylinder. *J of Comp Phys*. 1997. 134:306-315.

He X, Luo L, and Dembo M. Some Progress in Lattice Boltzmann Method. Part I. Nonuniform Mesh Grids. *J Comp Phys*. 1996. 129:357-363.

Heath C, and Magari S. Mini-review: Mechanical Factors Affecting Cartilage Regeneration in vitro. *Biotechnol Bioeng*. 1996. 50(4):430-437.

Hu HH, Joseph DD, and Crochet MJ. Direct Simulation of Fluid Particle Motions. *Theor Comp Fluid Dyn.* 1992. 3:285-306.

Hu HH, Patankar NA, and Zhu MY. Direct Numerical Simulations of Fluid-Solid Systems using Arbitrary Lagrangian-Eulerian Technique. *J Comp Phys.* 2001. 169:624-651.

Inamuro T, Yoshino M, and Ogino F. A Non-Slip Boundary Condition for Lattice Boltzmann Simulations. *Phys Fluids*. 1995 Dec. 7(12):2928-2930.

Inamuro T, Yoshino M, and Ogino F. Lattice Boltzmann Simulation of flows in a Three-Dimensional Porous Structure. *Int J Numer Meth Fluids*. 1999. 29:737-748.

Johnson AA, and Tezduyar TE. 3D Simulation of Fluid-Particle Interactions with the Number of Particles Reaching 100. *Comp Meth Appli Mech Eng*. 1997. 145:301-321.

Kandhai D, Koponen A, Hoekstra AG, Kataja M, Timonen J, and Sloot PMA. Lattice-Boltzmann Hydrodynamics on Parallel Systems. *Comp Phys Comm*. 1998. 111:14-26.

Kandhai D, Koponen A, Hoekstra A, Kataja M, Timonen J, and Sloot PMA. Implementation Aspects of 3D Lattice-BGK: Boundaries, Accuracy, and a New Fast Relaxation Method. *J Comp Phys*. 1999. 150:482-501.

Kandhai D, Soll W, Chen S, Hoedstra A, and Sloot P. Finite-Difference Lattice-BGK Methods on Nested Grids. *Comp Phys Comm*. 2000. 129:100-109.

Koponen A, Kataja M, Timonen J, and Kandhai D. Simulations of Single-Fluid Flow in Porous Media. *Int J of Mod Phys C*. 1998. 9(8):1505-21.

Ladd AJC. Numerical simulations of particulate suspensions via a discretized Boltzmann equation Part I. Theoretical foundation. *J Fluid Mech*. 1994a 271:285-309.

Ladd AJC. Numerical simulations of particulate suspensions via a discretized Boltzmann equation Part 2. Numerical Results. *J Fluid Mech*. 1994b 271:311-339.

Ladd AJC, and Verberg R. Lattice-Boltzmann Simulations of Particle-Fluid Suspensions. *J Stat Phys*. 2001. 104(5/6): 1191-1251.

Lai YG, Lin CL, and Huang J. Accuracy and Efficiency Study of Lattice Boltzmann Method for Steady-State Flow Simulations. *Numerical Heat Transfer B*. 2001. 39:21-43.

Lallemand P, and Luo LS. Theory of the Lattice Boltzmann Method: Dispersion, Dissipation, Isotropy, Galilean Invariance, and Stability. *Phys Rev E*. 2000. 61(6):6546-6562.

LeBaron RG, and Athanasiou KA. Ex Vivo Synthesis of Articular Cartilage. *Biomaterials*. 2000. 21(24):2575-2587.

Levesque MJ, Nerem RM. The Elongation and Orientation of Cultured Endothelial Cells in Response to Shear Stress. *J Biomech Eng*. 1985. 107(4):341-47.

Lin CL, and Lai YG. Lattice Boltzmann Method on Composite Grid. *Phys Rev E*. 2000. 62:2219-2225.

Ma PX, Langer R. Morphology and Mechanical Function of Long-Term *In Vitro* Engineered Cartilage. *J Biomed Mater Res*. 1999. 44(2):217-21.

Maier RS, and Bernard RS. Accuracy of the Lattice-Boltzmann Method. *Int J of Mod Phys C*. 1997. 8(4):747-752.

Maier RS, Bernard RS, and Grunau DW. Boundary Conditions for the Lattice Boltzmann Method. *Phys Fluids*. 1996. 8(7):1788-1801.

Malek AM, and Izumo S. Control of Endothelial Cell Gene Expression by Flow. *J Biomech*. 1995. 28(12):1515-28.

Martin I, Vunjak-Novakovic G, Yang J, Langer R, Freed LE. Mammalian Chondrocytes Expanded in the Presence of Fibroblast Growth Factor 2 Maintain the Ability to Differentiate and Regenerate Three-Dimensional Cartilaginous Tissue. *Exp Cell Research*. 1999. 253:681-688.

Mei R, Luo LS, and Shyy W. An Accurate Curved Boundary Treatment in the Lattice Boltzmann Method. *J Comp Phys*. 1999. 155:307-330.

Mei R, Shyy W, Yu D, and Luo L. Lattice Boltzmann Method for 3-D Flows with Curved Boundary. *J Comp Phys*. 2000. 161:680-699.

Mei R, Yu D, Shyy W, and Luo L. Force Evaluation in the Lattice Boltzmann Method Involving Curved Geometry. *Phys Rev E*. 2002. 65:041203.

Miyamura A, Iwasaki S, and Ishii T. Experiment Wall Correction Factors of Single Solid Spheres in Triangular and Square Cylinders, and Parallel Plates. *Int J Multiphase Flow*. 1981. 7(1):41-46.

Mow VC, Ratcliffe A. Structure and Function of Articular Cartilage and Meniscus. Basic Orthopaedic Biomechanics, 2nd ed. 1997. Lippincott-Raven Publishers, Philadelphia, PA. 113-178.

Neitzel GP, Nerem RM, Sambanis A, Smith MK, Wick TM, Brown JB, Hunter C, Jovanovic I, Malaviya P, Saini S, and Tan S. Cell Function and Tissue Growth in Bioreactors: Fluid-Mechanical and Chemical Environments. *J Japan Soc Microgravity Appl.* 1998. 15(II):602-607.

Nguyen NG, and Ladd AJC. Lubrication Corrections for Lattice-Boltzmann Simulations of Particle Suspensions. *Phys Rev E*. 2002. 66:046708.

Noble DR, Chen S, Geogiadis JG, and Buckius RO. A Consistent Hydrodynamic Boundary Condition for the Lattice Boltzmann Method. *Phys Fluids*. 1995. 7(1):203-209.

Obradovic B, Carrier RL, Vunjak-Novakovic G, Freed LE. Gas Exchange is Essential for Bioreactor Cultivation of Tissue Engineered Cartilage. *Biotechnol Bioeng*. 1999 Apr 20. 63(2):197-205.

Pei M, Solchaga LA, Seidel J, Zeng L, Vunjak-Novakovic G, Caplan AI, and Freed LE. Bioreactors Mediate the Effectiveness of Tissue Engineering Scaffolds. *FASEB*. 2002. 16:1691-1694.

Prosperetti A, and Tryggvason G. Appendix 3: Report of Study Group on Computational Physics. *Int J of MultiPhase Flow*. 2003. 29:1089-1099.

Qi D. Lattice-Boltzmann Simulations of Particles in Non-Zero-Reynolds-Number Flows. *J Fluid Mech*. 1999. 385:41-62.

Qi D. Lattice-Boltzmann Simulations of Fluidization of Rectangular Particles. *Int J of Multiphase Flow*. 2000. 26:421-433.

Qian YH. Simulating Thermohydrodynamcis with Lattice BGK Models. *J Sci Comp*. 1993. 8:231-41.

Qian YH, d'Humieres D, and Lallemand P. Lattice BGK Models for Navier-Stokes Equation. *Europhys Lett*.1992.17:479-484.

Roache PJ. Verification and Validation in Computational Science and Engineering. 1998. Hermosa Publishers, Albuquerque, NM.

Rosenhead L. Laminar Boundary Layers. 1963. Oxford University Press, London, UK.

Rothman DH, and Zaleski S. Lattice-Gas Models of Phase Separation: Interfaces, Phase Transistions, and Multiphase Flow. *Rev Mod Phys*. 1994. 66(4):1417-1479.

Satofuka N, and Nishioka T. Parallelization of Lattice Boltzmann Method for Incompressible Flow Computations. *Comp Mech*. 1999. 23:164-171.

Schwarz RP, Goodwin TJ, Wolf DA. Cell Culture for Three-Dimensional Modeling in Rotating-Wall Vessels: An Application of Simulated Microgravity. *J Tiss Cult Meth*. 1992 14:51-58.

Sethian JA. Evolution, Implementation, and Application of Level Set and Fast Marching Methods for Advancing Fronts. *J Comp Phys*. 2001. 169:503-555.

Slater JW. NPARC Alliance CFD Verification and Validation Web Site. http://www.grc.nasa.gov/WWW/wind/valid/homepage.html.

Smith RL, Donlon BS, Gupta MK, Mohtai M, Das P, Carter DR, Cooke J, Gibbons G, Hutchinson N, Schurman DJ. Effects of Fluid-Induced Shear on Articular Chondrocyte Morphology and Metabolism In Vitro. *J Orthop Res*. 1995. 13(6):824-31.

Spaulding GF, Jessup JM, Goodwin TJ. Advances in Cellular Construction. *J Cell Biochem*. 1993. 51(3):249-51.

Succi S, Karlin I, and Chen H. Colloquium: Role of the H Theorem in Lattice Boltzmann Hydrodynamic Simulations. *Rev Mod Phys*. 2002. 74:1203-1220.

Sucosky P, Osorio DF, Brown JB, and Neitzel GP. Fluid Mechanics of a Spinner-Flask Bioreactor. *Biotechnol and Bioeng*. 2004. 85(1):34-46.

Takagi S, Matsumoto Y, and Huang H. Numerical Analysis of a Single Rising Bubble using Boundary-Fitted Coordinate System. *JSME Int J B*. 40:42-50.

Takaisi Y. The Drag on a Circular Cylinder Moving with Low Speeds in a Viscous Liquid Between Two Parallel Plates. *J Phys Soc Japan*. 1955. 10:685-693.

Tryggvason G, Bunner B, Esmaeeli A, Juric D, Al-Rawahi N, Tauber W, Han J, Nas S, and Jan YJ. A Front Tracking Method for the Computations of Multiphase Flow. *J Comp Phys*. 2001. 169:708-759.

Ubertini S, Bella G, and Succi S. Lattice Boltzmann Method on Unstructured Grids: Further Developments. *Phys Rev E*. 2003. 68:016701.

Ujita H, Nagata S, Akiyama M, and Naitoh M. Development of LGA & LBE 2D Parallel Programs. *Int J of Mod Phys C*. 1998. 9(8):1203-1220.

van Wachem BGM, and Almstedt AE. Methods for Multiphase Computational Fluid Dynamics. *Chem Eng J*. 2003. 96:81-98.

Verberg R, and Ladd AJC. Lattice-Boltzmann Model with Sub-Grid-Scale Boundary Conditions. *Phys Rev Letters*. 2000. 84(10):2148-2151.

Verberg R, and Ladd AJC. Accuracy and Stability of a Lattice-Boltzmann Model with Subgrid Scale Boundary Conditions. *Phys Rev E*. 2002. 65:016701.

Wereley ST, and Lueptow RM. Inertial Particle Motion in a Taylor Couette Rotating Filter. *Phys Fluids*. 1999. 11(2):325-333.

White CM. The Drag of Cylinders in Fluids at Slow Speeds. *Proc Roy Soc London A*. 1946. 186:472-479.

Williams KA, Saini S, and Wick TM. Computational Fluid Dynamics Modeling of Steady-State Momentum and Mass Transport in a Bioreactor for Cartilage Tissue Engineering. *Biotechnol Prog*. 2002. 18:951-963.

Witkin A, and Baraff D. Physically Based Modeling: Principles and Practice. SIGGRAPH ’97 Course Notes, ACM SIGGRAPH, 1997.

Wolf DA, and Schwarz RP. Analysis of Gravity-Induced Particle Motion and Fluid Perfusion Flow in the NASA-Designed Rotating Zero-Head-Space Tissue Culture Vessel. NASA Tech Paper. 1991. 3143:1-12.

Yu D, Mei R, Luo LS, and Shyy W. Viscous Flow Computations with the Method of Lattice Boltzmann Equation. *Prog in Aero Sciences*. 2003. 39:329-367.

Yu D, Mei R, and Shyy W. A Multi-Block Lattice Boltzmann Method for Viscous Fluid Flows. *Int J Num Methods in Fluids*. 2002. 39:99-120.

Zou Q, and He X. On Pressure and Velocity Boundary Conditions for the Lattice Boltzmann BGK Model. *Phys Fluids*. 1997. 9(6):1591-1598.

Zou Q, Hou S, and Doolen GD. Analytical Solutions of the Lattice Boltzmann BGK Model. *J Stat Phys*. 1995. 81:319-334.

Vita

Marnico D. Deladisma was born in Chicago, Illinois in 1976 to Dr. Marconi R. Deladisma and Adelfa Deladisma. He has two older siblings, a brother Marconi and a sister Adeline. After moving several times, since his father served in the U.S. Air Force, his family settled down in upstate New York. Marnico attended the Georgia Institute of Technology for his undergraduate education and received his Bachelors in Mechanical Engineering in 1999. He then went on to pursue his doctorate at Georgia Tech, earning a Master of Science and Master of Business Administration along the way. He has been happily married to his lovely wife, Kah Mun, since 2002, and enjoys being involved in his home church in Atlanta, Georgia.